

(43) 国際公開日
2004年7月8日 (08.07.2004)

PCT

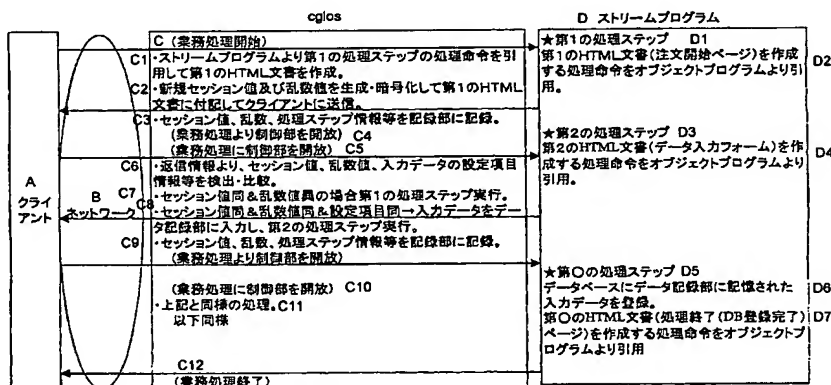
(10) 国際公開番号
WO 2004/057483 A1

- (51) 国際特許分類: G06F 15/00 (71) 出願人 および
(21) 国際出願番号: PCT/JP2003/016231 (72) 発明者: 澤田 研一 (SAWADA, Kenichi) [JP/JP]; 〒108-0072 東京都港区白金1丁目25-24-305 Tokyo (JP).
(22) 国際出願日: 2003年12月18日 (18.12.2003) (74) 代理人: 西脇 民雄, 外 (NISHIWAKI, Tamio et al.); 〒104-0061 東京都中央区銀座7丁目9番15号 銀座ガスホールビル3階 Tokyo (JP).
(25) 国際出願の言語: 日本語 (81) 指定国 (国内): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD,
(26) 国際公開の言語: 日本語
(30) 優先権データ: 特願 2002-370953
2002年12月20日 (20.12.2002) JP

[続葉有]

(54) Title: WEB APPLICATION SERVER SUPPORT PROGRAM AND SERVER

(54) 発明の名称: ウェブアプリケーションサーバ用支援プログラム及びサーバ



- A...CLIENT
B...NETWORK
C...(BUSINESS PROCESSING START)
C1...A FIRST HTML DOCUMENT IS CREATED BY CITING A PROCESSING INSTRUCTION OF FIRST PROCESSING STEP FROM STREAM PROGRAM.
C2...A NEW SESSION VALUE AND A RANDOM NUMBER ARE GENERATED, ENCRYPTED, AND ADDED TO THE FIRST HTML, WHICH IS SENT TO CLIENT.
C3...SESSION VALUE, RANDOM NUMBER, AND PROCESSING STEP INFORMATION ARE RECORDED IN THE RECORDING SECTION.
C4...(CONTROL SECTION IS RELEASED FROM BUSINESS PROCESSING.)
C5...(CONTROL SECTION IS RELEASED FOR BUSINESS PROCESSING.)
C6...SESSION VALUE, RANDOM NUMBER, AND INPUT DATA SET ITEM INFORMATION ARE DETECTED FROM RETURN INFORMATION AND COMPARED.
C7...IF SESSION VALUES ARE IDENTICAL AND THE RANDOM NUMBERS ARE DIFFERENT, FIRST PROCESSING STEP IS EXECUTED.
C8...IF SESSION VALUES ARE IDENTICAL, RANDOM NUMBERS ARE IDENTICAL, AND SET ITEMS ARE IDENTICAL, THEN THE INPUT DATA IS INPUT TO THE DATA RECORDING SECTION AND SECOND PROCESSING STEP IS EXECUTED.
C9...SESSION VALUE, RANDOM NUMBER, AND PROCESSING STEP INFORMATION ARE RECORDED IN THE RECORDING SECTION.
C10...SAME AS ABOVE
C11...AND SO FORTH
C12...(BUSINESS PROCESSING END)
D...STREAM PROGRAM
D1...FIRST PROCESSING STEP
D2...QUOTE PROCESSING INSTRUCTION FOR CREATING THE FIRST HTML DOCUMENT (ORDER START PAGE) FROM OBJECT PROGRAM.
D3...SECOND PROCESSING STEP
D4...QUOTE PROCESSING INSTRUCTION FOR CREATING THE SECOND HTML DOCUMENT (DATA INPUT FORM) FROM OBJECT PROGRAM.
D5...N-TH PROCESSING STEP
D6...REGISTER INPUT DATA STORED IN DATA RECORDING SECTION, IN DATABASE
D7...QUOTE PROCESSING INSTRUCTION FOR CREATING THE N-TH HTML DOCUMENT (PROCESSING END (DB REGISTRATION COMPLETE) PAGE) FROM OBJECT PROGRAM

(57) Abstract: A Web application server support program enables a user to completely control the execution order of the processing flow including transitions of a plurality of screens in the Web application at the server side and describe the processing flow as "one-unit executable program code". Moreover, by using the Web application server support program, it is possible to record all the items of the input form contained in the HTML document transmitted to a client as a form ledger in the server so that when input data corresponding to the input form is received from a client, it is compared/checked with the item on the form ledger, thereby automatically discarding an unauthorized input without depending on the application program.

[続葉有]



SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG,
US, UZ, VC, VN, YU, ZA, ZM, ZW.

TR), OAPI 特許 (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,
GW, ML, MR, NE, SN, TD, TG).

(84) 指定国 (広域): ARIPO 特許 (BW, GH, GM, KE, LS,
MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア特
許 (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッ
パ特許 (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,
FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK,

添付公開書類:

— 国際調査報告書

2文字コード及び他の略語については、定期発行される
各PCTガゼットの巻頭に掲載されている「コードと略語
のガイダンスノート」を参照。

(57) 要約: 本発明に係るウェブアプリケーションサーバ用支援プログラムを用いることによって、ウェブアプリケーションにおける複数画面の遷移を含む処理フローの実行処理順序をサーバ側で完全に制御することができ、当該処理フローを「一単位の実行可能なプログラムコード」として記述することが可能となる。また、ウェブアプリケーションサーバ用支援プログラムを用いることによって、クライアントに送信するHTML文書に含まれる入力フォームの各項目等を、サーバにフォーム台帳として逐一記録させておき、クライアントから当該入力フォームに対応する入力データを受け取る際にフォーム台帳の項目と比較・検証させて、アプリケーションプログラムに依存することなく不正な入力を自動的に破棄等させることが可能となる。

明細書

ウェブアプリケーションサーバ用支援プログラム及びサーバ

【技術分野】

本発明は、クライアントにウェブ情報を送信するとともに、ウェブ情報に対するクライアントからの返信情報を受信して、返信情報に基づく一連の処理を実行するサーバとこのサーバに用いられる支援プログラムに関する。

【背景技術】

インターネット等のネットワークに接続されたウェブサーバに対して、クライアントからウェブ情報（例えばHTML文書）の送信要求があった場合には、図22に示すように、ウェブサーバ30は、要求対象となるウェブ情報31をURLによって特定してクライアント32に送信することによって、クライアント32の要求に応じた情報提供を行っている。

今日では、クライアント32による送信要求に応じてウェブ情報31をただ単に返信するだけではなく、図23に示すように、クライアント32からの要求に応じてある一連の処理を行うウェブアプリケーションサーバ33が数多く用いられている。ウェブアプリケーションサーバ33は、いわゆるフロントエンド（クライアント32）とバックエンド（データベース管理システム等）との中間に位置する一種の支援プログラム（ミドルウェア）を備えており、この支援プログラムは、一連の処理を行うために必要な処理環境を整えて各種処理の流れを効率よく制御する役割を果たしている。また、ウェブアプリケーションサーバ33を用いることによって、ブラウザを備えているクライアント32であれば、ほとんどの処理をウェブアプリケーションサーバ33で処理させることができるので、クライアント32のハードウェア能力に依存せずに処理

を行うことが可能となっている（例えば、日経ＢＰ社出版局編「デジタル用語辞典２００２－２００３年版」日経ＢＰ社発行、２００２年３月１８日第３版、４３１ページ参照）。

ところで、既存のウェブアプリケーションサーバ３３では、一連の処理に必要なＨＴＭＬ文書を生成するためのプログラム（以下、ビューとする。）が予め用意されており、ウェブアプリケーションサーバ３３は、クライアント３２のＨＴＭＬ文書の送信要求に応じて順次処理に必要なＨＴＭＬ文書を生成してクライアント３２に送信し、そのＨＴＭＬ文書の入力フォームに基づいて入力された入力データをクライアント３２から受信することによって、一連の処理に必要なデータを取得する。

例えば、図２４に示すような商品の発注処理を行うウェブアプリケーションサーバ３３においては、クライアント３２で注文入力を行うための注文入力画面を生成するビュー（以下、「ビューＡ」とする。）と、注文入力画面の入力フォームに対するクライアント３２からの注文データに応じてその注文データの確認をクライアント３２で行ってもらうための注文確認画面を生成するビュー（以下、「ビューＢ」とする。）と、注文確認画面における注文了解データに応じて注文処理が完了した旨の表示を行う注文完了画面を生成するビュー（以下、「ビューＣ」とする。）とが予め用意されている。ウェブアプリケーションサーバ３３は、注文データや注文了解データをクライアント３２から受信すると、注文内容（例えば、銘柄、数量、合計金額、発送先等）をデータベース３４に記憶し、データベース３４に記憶された注文内容に基づいて商品発送処理等を行っている。

【発明の開示】

【発明が解決しようとする課題】

しかしながら、ウェブアプリケーションサーバ３４では、ビューＡ～

ビューCがネットワークを介していつでもアクセス可能な状態となっているので、クライアント32は発注処理の流れに関係なくビューA～ビューCにアクセスすることができる。図24に示した発注処理において処理がビューAからビューCまでの手順に従って実際に行われているか否かは、クライアント32の送信要求に委ねられている。

このため、図25に示すように、ビューAに対するクライアント32からの応答に対して次のHTML文書の送信要求先がビューBとなるようにビューAには次の送信要求先が記憶されているが、その送信要求先がビューBになるか否かはクライアント32の送信要求次第であり、ビューAの次にビューBが送信される予定となっても、場合によってはクライアント32がビューBではなくビューCの送信要求を行ってしまい、ウェブアプリケーションサーバ33がビューBにより生成されるHTML文書ではなくビューCにより生成されたHTML文書をクライアント32に送信してしまうおそれがある。

また、図26に示すように、クライアント32においてビューBの注文確認用のHTML文書が送信され、その後にビューCにより生成された発注処理が完了した旨のHTML文書が送信され発注処理が完了した後に、クライアント32はブラウザの「戻る」ボタンを押すことによってキャッシュ内に記憶されていたビューBのHTML文書の画面を再度表示させることができる。クライアント32は、このようにキャッシュ内に記憶されていたビューBの画面に従って再度ウェブアプリケーションサーバ33に注文了解データを返信してしまうおそれがあり、再度注文了解データをウェブアプリケーションサーバ33に返信した場合には二重に注文処理を行ってしまうという問題がある。

さらに、図27に示すように、クライアント32がビューAに対応する注文データをウェブアプリケーションサーバ33に返信する前に、悪

意のある第三のクライアント 35 がクライアント 32 になりすまして他の注文データを送信してしまうおそれがある。この場合、従来のウェブアプリケーションサーバ 33 では、送信された注文データが、どのクライアントから送信されたどの HTML 文書に基づく返信であるかを判断することができないので、受信し注文データをクライアント 32 より送信された注文データであると判断して発注処理を進めてしまう場合があり、クライアント 32 とウェブアプリケーションサーバ 33 との間で安全な発注処理を行うことが困難であるという問題があった。

また、一般的な HTML 文書においては、ビュー A～ビュー C のそれぞれのビュー内に、クライアント 32 が入力する必要のある入力項目が直接記述されているので、処理の変更等により入力項目が増減した場合には、各ビューの項目をそれぞれのビュー毎に変更しなければならず、変更処理に多大な労力を要するという問題があった。

本発明は、上記の問題に鑑みて為されたものであり、クライアントにウェブ情報を送信するとともに、ウェブ情報に対するクライアントからの返信情報を受信して、返信情報に基づく一連の処理を実行するサーバにおいて、サーバが一連の処理の処理過程を把握することによって処理手順に沿わない処理がなされることを防止することができ、さらに、一連の処理に用いられるウェブ情報の内容を容易に変更することが可能なサーバ及びそのサーバに用いられる支援プログラムを提供することを課題とする。

【課題を解決するための手段】

上記課題を解決するために、本発明は、通信手段を用いてクライアントにウェブ情報を送信し、該通信手段において前記ウェブ情報に対する前記クライアントからの返信情報を受信して該返信情報を記憶手段に記憶させ、前記記憶手段に記憶される一連の処理の処理ステップに従って、

前記返信情報に基づく処理を実行するウェブアプリケーションサーバの制御部に、前記クライアントを特定するためのクライアント特定情報と前記ウェブ情報を特定するためのウェブ特定情報とを当該ウェブ情報に付加させる情報付加処理と、前記クライアント特定情報と前記ウェブ特定情報とが付加された前記ウェブ情報を特定の前記処理ステップにおいて前記通信手段を用いて前記クライアントに送信させる情報送信処理と、前記ウェブ情報に付加された前記クライアント特定情報及び前記ウェブ特定情報と、前記ウェブ情報が前記クライアントに送信されたときの前記処理ステップに関する処理情報とを、前記ウェブ情報が前記クライアントに送信される度に前記記憶手段に記憶させる情報記憶処理と、前記クライアントより返信される返信情報を、前記通信手段において受信させる情報受信処理と、受信された返信情報にクライアント特定情報とウェブ特定情報とが含まれていないかを検出させる返信情報検出処理と、前記返信情報よりクライアント特定情報及びウェブ特定情報が検出され、かつ、検出された前記クライアント特定情報及び前記ウェブ特定情報が前記記憶手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一である場合に、前記記憶手段に記憶された前記処理情報より求められる処理ステップの次の処理ステップを前記記憶手段より求めて実行させ、前記返信情報よりクライアント特定情報及びウェブ特定情報が検出され、かつ、検出された前記クライアント特定情報と前記記憶手段に記憶される前記クライアント特定情報とは同一であるが、前記返信情報より検出されたウェブ特定情報と前記記憶手段に記憶される前記ウェブ特定情報とが異なる場合には、前記記憶手段に記憶された前記処理情報より求められる処理ステップを再度実行させる業務実行処理とを実行させるウェブアプリケーションサーバ用支援プログラムであることを特徴とする。

また、前記記憶手段が、メンバ変数の設定項目別の作業領域として前記一連の処理において使用されるデータ宣言部と、前記一連の処理において前記返信情報に含まれるデータを前記設定項目毎に記憶するデータ記憶部とを備え、前記制御部に対して前記ウェブ情報に前記クライアントにデータを入力させるためのデータ入力フォームが含まれている場合には、当該ウェブ情報を前記クライアントに送信する際に前記データ入力フォームにより入力される入力データの設定項目の種類を前記データ宣言部より判別し、前記情報記憶処理において前記クライアント特定情報、前記ウェブ特定情報、前記処理情報とともに前記記憶手段に記憶させ、前記返信情報より検出されたクライアント特定情報及びウェブ特定情報が前記記憶手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一であって、前記返信情報より検出された入力データの設定項目の種類と前記記憶手段に記憶される入力データの設定項目の種類とが同一である場合には、前記業務実行処理において、検出された前記入力データを設定項目毎に前記データ記憶部に記憶させる様にしてもよい。

さらに、前記ウェブ情報に前記データ入力フォームが複数含まれている場合であって、前記データ入力フォームに従って入力される入力データの設定項目の種類が重複する場合には、前記制御部に、設定項目の種類が重複する入力データの識別を行うための識別子を前記データ入力フォーム毎に付して前記ウェブ情報に含ませる識別子追加処理を実行させるとともに、前記情報記憶処理において、前記クライアント特定情報、前記ウェブ特定情報、前記処理情報とともに、前記データ入力フォームにおいて入力される入力データの設定項目の種類を前記識別子に関連付けて前記記憶手段に記憶させ、前記業務実行処理において、前記返信情報より検出されたクライアント特定情報及びウェブ特定情報が前記記憶

手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一であって、前記返信情報より検出される前記入力データの設定項目の種類と前記記憶手段に記憶される設定項目の種類とが同一識別子毎に一致する場合には、検出された前記入力データの設定項目を前記記憶手段に記憶させておいた識別子に基づいて区別して設定項目毎に前記データ記憶部に記憶させる様にしてもよい。

また、本発明は、クライアントにデータを入力させるためのデータ入力フォームが含まれるウェブ情報を通信装置を用いて前記クライアントに送信し、該通信装置において前記データ入力フォームに従って入力された入力データを返信情報として前記クライアントから受信し、メンバ変数の設定項目別の作業領域として使用されるデータ宣言部と、前記返信情報に含まれるデータを前記設定項目毎に記憶するデータ記憶部とを備える記憶装置に前記入力データを記憶させるウェブアプリケーションサーバの制御部に対して、当該ウェブ情報に前記クライアントにデータを入力させるためのデータ入力フォームを付加させる情報付加処理と、前記データ入力フォームが付加された前記ウェブ情報を前記通信手段を用いて前記クライアントに送信させる情報送信処理と、前記データ入力フォームにより入力される入力データの設定項目の種類を前記記憶手段に記憶される前記データ宣言部より判別して、前記ウェブ情報が前記クライアントに送信される度に前記記憶手段に記憶させる情報記憶処理と、前記クライアントより返信される返信情報を、前記通信手段において受信させる情報受信処理と、受信された返信情報に前記入力データが含まれていないかを検出させる返信情報検出処理と、前記返信情報より前記入力データが検出され、かつ検出された入力データの設定項目の種類と、前記記憶手段に記憶される設定項目の種類とが同一である場合には、検出された前記入力データを設定項目毎に前記データ記憶部に記憶させる

業務実行処理とを実行させるウェブアプリケーションサーバ用支援プログラムであることを特徴とする。

このウェブアプリケーションサーバ用支援プログラムにおいて、前記ウェブ情報に前記データ入力フォームが複数含まれている場合であって、前記データ入力フォームに従って入力される入力データの設定項目の種類が重複する場合には、前記制御部に、設定項目の種類が重複する入力データの識別を行うための識別子を前記データ入力フォーム毎に付して前記ウェブ情報に含ませる識別子追加処理を実行させるとともに、前記情報記憶処理において、前記データ入力フォームにおいて入力される入力データの設定項目の種類を前記識別子に関連付けて前記記憶手段に記憶させ、前記業務実行処理において、前記返信情報より検出される前記入力データの設定項目の種類と前記記憶手段に記憶される設定項目の種類とが同一識別子毎に一致する場合には、検出された前記入力データの設定項目を前記記憶手段に記憶させた識別子に基づいて区別して設定項目毎に前記データ記憶部に記憶させる様にしてもよい。

さらに望ましくは、上述したウェブアプリケーションサーバ用支援プログラムにおいて、前記情報記憶処理の後に、前記情報受信処理を行うために必要な情報を前記記憶手段に退避させて前記制御部に他の処理を実行させ、前記情報受信処理により前記返信情報を受信したときに、前記退避された情報を復帰させて前記制御部に前記返信情報検出処理を再開させることで前記ウェブアプリケーションサーバにおいて並列処理を実現させるようにしても良い。

また、上述したウェブアプリケーションサーバ用支援プログラムにより実現される機能を備えたウェブアプリケーションプログラムであってもよく、さらに、ウェブアプリケーションサーバ用支援プログラムにより実現される機能を備えたオペレーティングシステムであってもよい。

さらに本発明は、通信手段を用いてクライアントにウェブ情報を送信し、該通信手段において前記ウェブ情報に対する前記クライアントからの返信情報を受信して該返信情報を記憶手段に記憶させ、前記記憶手段に記憶される前記一連の処理の処理ステップに従って、前記返信情報に基づく処理を制御部が実行するサーバにおいて、前記クライアントを特定するためのクライアント特定情報と前記ウェブ情報を特定するためのウェブ特定情報とを当該ウェブ情報に付加させる情報付加処理と、前記クライアント特定情報と前記ウェブ特定情報とが付加された前記ウェブ情報を特定の前記処理ステップにおいて前記通信手段を用いて前記クライアントに送信する情報送信処理と、前記ウェブ情報に付加された前記クライアント特定情報及び前記ウェブ特定情報と、前記ウェブ情報が前記クライアントに送信されたときの前記処理ステップに関する処理情報とを、前記ウェブ情報が前記クライアントに送信される度に前記記憶手段に記憶させる情報記憶処理と、前記クライアントより返信される返信情報を、前記通信手段に受信させる情報受信処理と、受信された返信情報にクライアント特定情報とウェブ特定情報とが含まれていないかを検出する返信情報検出処理と、前記返信情報よりクライアント特定情報及びウェブ特定情報が検出され、かつ、検出された前記クライアント特定情報及び前記ウェブ特定情報が前記記憶手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一である場合に、前記記憶手段に記憶された前記処理情報より求められる処理ステップの次の処理ステップを前記記憶手段より求めて実行し、前記返信情報よりクライアント特定情報及びウェブ特定情報が検出され、かつ、検出された前記クライアント特定情報と前記記憶手段に記憶される前記クライアント特定情報とは同一であるが、前記返信情報より検出されたウェブ特定情報と前記記憶手段に記憶される前記ウェブ特定情報とが異なる場合には、前記

記憶手段に記憶された前記処理情報より求められる処理ステップを再度実行する業務実行処理とを前記制御部が実行するサーバであることを特徴とする。

また、前記記憶手段が、メンバ変数の設定項目別の作業領域として前記一連の処理において使用されるデータ宣言部と、前記一連の処理において前記返信情報に含まれるデータを前記設定項目毎に記憶するデータ記憶部とを備え、前記制御部は、前記ウェブ情報に前記クライアントにデータを入力させるためのデータ入力フォームが含まれている場合に、当該ウェブ情報を前記クライアントに送信する際に前記データ入力フォームにより入力される入力データの設定項目の種類を前記データ宣言部より判別し、前記情報記憶処理において前記クライアント特定情報、前記ウェブ特定情報、前記処理情報とともに前記記憶手段に記憶し、前記返信情報より検出されたクライアント特定情報及びウェブ特定情報が前記記憶手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一であって、前記返信情報より検出された入力データの設定項目の種類と前記記憶手段に記憶される入力データの設定項目の種類とが同一である場合には、前記業務実行処理において、検出された前記入力データを設定項目毎に前記データ記憶部に記憶してもよい。

さらに、前記ウェブ情報に前記データ入力フォームが複数含まれている場合であって、前記データ入力フォームに従って入力される入力データの設定項目の種類が重複する場合には、前記制御部は、設定項目の種類が重複する入力データの識別を行うための識別子を前記データ入力フォーム毎に付して前記ウェブ情報に含ませる識別子追加処理を行うとともに、前記情報記憶処理において、前記クライアント特定情報、前記ウェブ特定情報、前記処理情報とともに、前記データ入力フォームにおいて入力される入力データの設定項目の種類を前記識別子に関連付けて前

記記憶手段に記憶させ、前記業務実行処理において、前記返信情報より検出されたクライアント特定情報及びウェブ特定情報が前記記憶手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一であって、前記返信情報より検出される前記入力データの設定項目の種類と前記記憶手段に記憶される設定項目の種類とが同一識別子毎に一致する場合には、検出された前記入力データの設定項目を前記記憶手段に記憶させた識別子に基づいて区別して設定項目毎に前記データ記憶部に記憶させる様にしてもよい。

また、本発明は、クライアントにデータを入力させるためのデータ入力フォームが含まれるウェブ情報を通信手段を用いて前記クライアントに送信し、該通信手段において前記データ入力フォームに従って入力された入力データを返信情報として前記クライアントから受信し、メンバ変数の設定項目別の作業領域として使用されるデータ宣言部と、返信情報に含まれるデータを前記設定項目毎に記憶するデータ記憶部とを備える記憶手段に前記入力データを記憶させ制御部を有するサーバにおいて、当該ウェブ情報に前記クライアントにデータを入力させるためのデータ入力フォームを付加する情報付加処理と、前記データ入力フォームが付加された前記ウェブ情報を前記通信手段を用いて前記クライアントに送信する情報送信処理と、前記データ入力フォームにより入力される入力データの設定項目の種類を前記記憶手段に記憶される前記データ宣言部より判別して、前記ウェブ情報が前記クライアントに送信される度に前記記憶手段に記憶させる情報記憶処理と、前記クライアントより返信される返信情報を、前記通信手段に受信させる情報受信処理と、受信された返信情報に前記入力データが含まれていないかを検出する返信情報検出処理と、前記返信情報より前記入力データが検出され、かつ検出された入力データの設定項目の種類と、前記記憶手段に記憶される設定項目

の種類とが同一である場合には、検出された前記入力データを設定項目毎に前記データ記憶部に記憶する業務実行処理とを前記制御部が実行するサーバであることを特徴とする。

上述したサーバにおいて、前記ウェブ情報に前記データ入力フォームが複数含まれている場合であって、前記データ入力フォームに従って入力される入力データの設定項目の種類が重複する場合には、前記制御部は、設定項目の種類が重複する入力データの識別を行うための識別子を前記データ入力フォーム毎に付して前記ウェブ情報に含ませる識別子追加処理を行うとともに、前記情報記憶処理において、前記データ入力フォームにおいて入力される入力データの設定項目の種類を前記識別子に関連付けて前記記憶手段に記憶させ、前記業務実行処理において、前記返信情報より検出される前記入力データの設定項目の種類と前記記憶手段に記憶される設定項目の種類とが同一識別子毎に一致する場合には、検出された前記入力データの設定項目を前記記憶手段に記憶させた識別子に基づいて区別して設定項目毎に前記データ記憶部に記憶させてもよい。

【発明の効果】

以上説明したように本発明を用いることによって、サーバが実行する一連の処理をサーバでその処理ステップの処理状況を確認しながら実行することが可能となる。

また、本発明では、返信情報に含まれる入力データの設定項目の種類が、送信したウェブ情報の入力フォームにおいて入力可能な設定項目の種類に対応するものであるか否かを制御部が判断して、入力データの設定項目と記憶された設定項目とが一致する場合にはその入力データをデータ宣言部に関連付けて自動的にデータ記憶部に記憶させることができるので、意図しない入力データがクライアントからの返信情報として

データ記憶部に記憶されてその後の処理に反映されることを防止することができる。

さらに、ウェブ情報の送信時及び返信情報の受信時にのみ、記憶部に記憶された必要なデータを読み出して（復帰させて）制御部に必要な処理を行わせることができるので、制御部は複数の一連の処理を同時に実行するマルチタスク環境を容易に実現することが可能となる。

【図面の簡単な説明】

【図 1】

本発明に係るサーバのソフトウェア構成を示したブロック図である。

【図 2】

本発明に係るサーバのハードウェア構成を示したブロック図である。

【図 3】

本発明に係るサーバとクライアントとのデータの送受信関係を示したデータフロー図である。

【図 4】

牡蛎料理の注文処理の処理手順を示したストリームプログラムのプログラムリストを示した図である。

【図 5】

牡蛎料理の注文処理に用いられるクラス定義プログラムのプログラムリストを示した第 1 の図である。

【図 6】

牡蛎料理の注文処理に用いられるクラス定義プログラムのプログラムリストを示した第 2 の図である。

【図 7】

牡蛎料理の注文処理に用いられる HTML 文書のデータ定義示した図である。

【図 8】

牡蛎料理の注文処理のステップ 2 において使用される HTML 文書の表示画面を示した第 1 の図である。

【図 9】

牡蛎料理の注文処理のステップ 2 において使用される HTML 文書の表示画面を示した第 2 の図である。

【図 10】

牡蛎料理の注文処理のステップ 3 において使用される HTML 文書の表示画面を示した第 1 の図である。

【図 11】

牡蛎料理の注文処理のステップ 3 において使用される HTML 文書の表示画面を示した第 2 の図である。

【図 12】

牡蛎料理の注文処理における調味料の種類に関するエラー表示が為された HTML 文書の表示画面を示した図である。

【図 13】

牡蛎料理の注文処理のステップ 5 において使用される HTML 文書の表示画面を示した図である。

【図 14】

牡蛎料理の注文処理において牡蛎の銘柄、牡蛎の調理方法、添える調味料、コメントの項目が 2 組表示された HTML 文書の表示画面を示した図である。

【図 15】

3 つのウィンドウを有する HTML 文書の表示画面の概略を示した図である。

【図 16】

3つのウィンドウを有するHTML文書を作成するためのストリームプログラムのプログラムリストを示した図である。

【図17】

3つのウィンドウを有するHTML文書を作成するために用いられるページ定義を示した図である。

【図18】

3つのウィンドウを有するHTML文書の表示画面を示した図である。

【図19】

識別子を用いて牡蛎の銘柄、コメントからなる複数組の設定項目を区別して記憶部に記憶させ、返信情報よりそれぞれの識別子に対応する入力データの設定項目を抽出してそれぞれデータ記憶部に記憶する処理を示した模式図である。

【図20】

ストリームプログラムの有無に拘わらず、設定項目を記憶部に記憶させ、返信情報より対応する設定項目を抽出してデータ記憶部に記憶する処理を示した模式図である。

【図21】

積算情報を時分情報に変換して記憶部に記憶させ、返信情報より対応する時分情報を抽出して積算情報に復元してデータ記憶部に記憶する処理を示した模式図である。

【図22】

ウェブサーバとクライアントとの間におけるウェブ情報の送受信方法を示した概略図である。

【図23】

従来のウェブアプリケーションサーバとクライアントとの間におけるウェブ情報の送受信方法を示した概略図である。

【図 2 4】

ウェブアプリケーションサーバとクライアントとの間における商品の発注処理の方法を示した概略図である。

【図 2 5】

ウェブアプリケーションサーバに対してクライアントが発注処理の流れに関係なくウェブ情報の送信要求を行ってしまう場合を示した図である。

【図 2 6】

ウェブアプリケーションサーバに対してクライアントが二重に発注処理をしてしまう場合を示した図である。

【図 2 7】

ウェブアプリケーションサーバに対して第三のクライアントが注文データを送信してしまう場合を示した図である。

【発明を実施するための最良の形態】

以下、本発明の実施の形態を、図面を用いて説明する。

図 1 は、本発明に係るウェブアプリケーションサーバのソフトウェア構成を示したブロック図である。このサーバ 1 は、Windows（登録商標）やLinux等のOS（operating system）2 と、OS 2 上で動作するcgios 3 と、cgios 3 に基づいて実行されるアプリケーションソフトウェア 4 とによって構成されている。

cgios 3 は、cgios 用に記載されたアプリケーションソフトウェア 4 を実行するミドルウェアであり、アプリケーションソフトウェア 4 を従来のOSであるOS 2 上で動かすためのエミュレータとしての役割を有している。また、cgios 3 は、ウェブサーバ用アプリケーションとしての機能及びCGI（Common Gateway Interface）機能に加えて、後述するcgios特有の機能を備えているので、ウェブア

アプリケーションサーバ用の支援ソフトウェア（ウェブアプリケーションサーバ用支援プログラム）としての役割も有している。この `cgi os 3` を用いることによって、`cgi os 3` において実現される処理を、プログラマーがアプリケーションソフトウェア 4 用のプログラムにおいて直接記述しなくても、`cgi os 3` が自動的に実行するので、プログラム作成労力の軽減を図ることが可能となる。

図 2 は、サーバ 1 のハードウェア構成の概略を示したブロック図である。サーバ 1 は、クライアント 7 との間で各種データ（情報）の送受信を行う送受信部 8 と、CPU からなる制御部 9 と、ハードディスクからなる記憶部 10 及びデータベース 11 とを備えている。

クライアント 7 は、パソコン、PDA (Personal Digital Assistance) 又は携帯電話等のブラウザを備えた通信機器である。送受信部 8 は NIC (Network Interface Card) 等により構成され、この送受信部 8 を用いることによって、サーバ 1 はネットワーク 12 を介してクライアント 7 との間でデータの送受信を行うことが可能となっている。ネットワーク 12 は外部に公開されるインターネット等のオープンなネットワークであっても、外部に公開されていない LAN 等のネットワークであってもよい。なお、図 2 においてはネットワーク 12 にはクライアント 7 が 1 台のみ接続されているが、複数接続されていてもかまわない。

記憶部 10 には、`cgi os 3` 用のプログラム言語により記述されたアプリケーションソフトウェア 4 が記録されている。アプリケーションソフトウェア 4 は、ストリームプログラムとクラス定義プログラムとがコンパイルされることにより生成される。サーバ 1 は、そのアプリケーションソフトウェア 4 に基づいて、例えばクライアントから商品の注文を受け付ける受付処理のような一連の窓口処理（フロントエンドに対す

るバックエンド側のインターフェースに関する処理)を実行する。

ここで、ストリームプログラムにはその窓口処理の全体的な流れが記述され、窓口処理の各段階における具体的な処理はクラス定義プログラムに記述されている。そして、このクラス定義プログラムがストリームプログラムに引用され、窓口処理が実行される。また、`cgi os 3`はクラス定義プログラムで使用されるメンバ変数をクラス定義プログラムのインスタンス（オブジェクト指向プログラム技術において一般的に使用されているインスタンスを意味し、クラスがオブジェクトの型を示すのに対して、インスタンスはオブジェクトの実体を示す。）ごとに管理する機能を有し、あるクラス定義プログラムのインスタンスからクライアント 7 に向けて送信されたメンバ変数が返信時にそのインスタンスのメンバ変数に戻るよう制御しているので（メンバ変数を発行するインスタンスと、メンバ変数を吸収するインスタンスとが同一）、ストリームプログラムがクラス定義プログラムに記載された処理命令を引用する場合には、ストリームプログラムにおいてメンバ変数を直接引数として記載する必要がない。したがって、引用される処理命令が異なるクラスに属するものであっても、また同一のクラスに属するものであっても、ストリームプログラムではそのクラスにおけるメンバ変数に左右されることなく処理手順を示すことが可能となっている。

また、記憶部 10 には、窓口処理における処理の段階を示す処理ステップ情報や、次述するセッション値（クライアント特定情報）及び乱数値（ウェブ特定情報）が記憶（退避）される。ここで、処理ステップ情報とは、より具体的には制御部 9 が処理を行う際に参照するプログラムカウンタのアドレス値を記憶した情報である。さらに、記憶部 10 には、窓口処理においてクライアント 7 からデータが入力された際に、そのときの処理に関わるインスタンスごとに入力データを設定項目として分け

て記憶するデータ記憶部が設けられている。

データベース 11 には、窓口処理において処理が確定した入力データが入力される。データベース 11 は、この点において、窓口処理において入力された入力データを一時的に保存するデータ記憶部と相違する。データベース 11 に記憶される設定項目は、クラス定義プログラムで宣言されるメンバ変数の設定項目と同期されており、クラス定義プログラムにおいて設定項目が変更されると、c g i o s 3 が自動的に同期される設定項目を変更する。

制御部 9 は、サーバ 1 を構成するコンピュータの CPU であり、アプリケーションソフトウェア 4 に従って各種処理を行うほか、c g i o s 3 のカーネル（c g i o s の基本機能を実現する部分）の指示に従って c g i o s 3 特有の機能を実現させるための処理を実行する。以下、c g i o s 3 の指示に基づいて制御部 9 によって実行される c g i o s 3 特有の機能を説明する。

図 3 は、サーバ 1 からクライアント 7 に送信される HTML 文書（ウェブ情報）及びクライアント 7 からサーバ 1 に返信される返信情報との送受信関係を示したデータフロー図である。

クライアント 7 によって取引開始ページの HTML 文書の送信要求（取引開始ページの URL へのアクセス要求）がなされた場合、制御部 9 は、セッション値と乱数値と生成する。ここで、セッション値とは、サーバ 1 が特定のクライアント 7 のために行う一連の窓口処理を識別するために用いられる値であり、制御部 9 は、このセッション値を用いることによってその窓口処理の内容の特定及びクライアント 7 の特定を行う。また、乱数値は、サーバ 1 がクライアント 7 に対して送信する HTML 文書を特定するために用いられるものである。なお、本実施例では HTML 文書の特定を行うために乱数値を用いることとしたが、HTML

L文書の特定を行うことが可能な識別情報としての機能を有するものであるならば乱数値に限定されるものではない。また、サーバ1は、ストリームプログラムに記述されている第1の処理ステップにおける処理命令をクラス定義プログラムから引用することによって第1の処理ステップに用いられるHTML文書（以下、第1のHTML文書とする。）を生成する。

その後、制御部9は、生成したセッション値と乱数値とを暗号化し、暗号化されたセッション値と乱数値とをHTMLのHIDDENパラメータを用いて第1のHTML文書に付加して送受信部よりクライアント7に送信する。また、制御部9は、第1のHTML文書の送信と同時に、セッション値、乱数値、処理ステップ情報を記憶部10に記憶（退避）し、さらに、第1のHTML文書にクライアント7にデータを入力させるための入力フォームが設けられている場合には、入力フォームにより入力することが可能な設定項目情報をインスタンス毎に記憶する。その後、制御部9は窓口処理から解放される。

サーバ1より送信された第1のHTML文書がクライアント7において受信されると、第1のHTML文書がクライアント7のウェブブラウザ画面上に表示される。クライアント7は、第1のHTML文書の入力フォームに従ってデータ（入力データ）を入力することができ、入力処理を行った後に「次へ」等のボタンをクリックすることによって、取引開始ページのURLへのアクセス要求を再度行うとともに、入力処理が行われた入力データと、暗号化されたセッション値及び乱数値とを返信情報としてサーバ1に返信する。なお、サーバ1のcgios3は、セッション値及び乱数値に基づいてクライアント及びHTML文書の特定を行っているため、クライアント7からサーバ1へのアクセス要求が行われるURLは、取引開始ページのURLだけである。この点において、

サーバ 1 は、窓口処理の処理内容毎に異なる URL をアクセス要求先として指定する従来のウェブアプリケーションサーバと相違する。

返信情報がサーバ 1 の送受信部 8 に返信されると、制御部 9 は、一連の窓口処理を再開する。制御部 9 では、返信情報に暗号化されたセッション値および乱数値が含まれているか否かの検出を行い、セッション値及び乱数値が含まれていない場合には、受信した返信情報はサーバ 1 が送信した HTML 文書とは無関係に返信された情報であると判断する。

返信情報より暗号化されたセッション値および乱数値が検出された場合には、制御部 9 は、暗号化されたセッション値及び乱数値を抽出して復号化するとともに、記憶部 10 に記憶されていたセッション値及び乱数値を読み出して、復号化されたセッション値及び乱数値を、記憶部 10 に記憶されているセッション値及び乱数値と比較する。制御部 9 は、復号化されたセッション値と記憶部 10 に記憶されているセッション値とが同一である場合には、HTML 文書を送信したクライアントと、返信情報を返信したクライアントとが同一クライアントであると判断する。一方、比較されるセッション値が異なる場合、制御部 9 はクライアントが異なるものと判断して受信した返信情報を破棄する。

比較されるセッション値が同一である場合、制御部 9 は、さらに復号化された乱数値と第 1 の HTML 文書を送信したときに記憶部 10 に記憶された乱数値とが同一であるか否かを比較し、乱数値が同一である場合には、第 1 の HTML 文書の入力フォームに従ってクライアント 7 が入力した入力データが返信情報として返信されたものであると判断し、乱数値が異なる場合には第 1 の HTML 文書とは異なる HTML 文書に基づく処理に対する返信情報であると判断する。

セッション値及び乱数値を比較することによって、制御部 9 が、受信した返信情報が第 1 の HTML 文書の入力フォームに基づく返信情報であ

ると判断した場合には、記憶部 10 に記憶されているクラス定義プログラムにおけるインスタンス毎の設定項目情報と、返信情報のインスタンス毎の入力データの設定項目情報とを比較し、記憶部 10 に記憶されている設定項目と一致する入力データを、そのインスタンスにおいて用いられる設定項目毎に区分けしてデータ記憶部に記憶する。このように、クライアント 7 によって入力された入力データをクラス定義プログラムに用いられるインスタンスのメンバ変数の設定項目毎にデータ記憶部に記憶させておくことによって、ストリームプログラムがその後の処理ステップにおいてクラス定義プログラムに記載された処理命令を引用する場合には、データ記憶部に記憶されたインスタンス毎の入力データを設定項目毎の既設値としてその後の処理に利用される。記憶部 10 に記憶されている設定項目と一致しない入力データは破棄される。その後、制御部 9 は、記憶部 10 に記憶された処理ステップ情報より、ストリームプログラムに示された一連の窓口処理においてどの処理ステップまで処理が終了しているかを判断し、判断される処理ステップの次の処理ステップ、つまり第 2 の処理ステップの処理を実行する。

第 2 の処理ステップにおいて、制御部 9 は、ストリームプログラムの第 2 の処理ステップで引用されている処理命令を実行することによって第 2 の処理ステップに用いられる HTML 文書（以下、第 2 の HTML 文書とする。）を生成する。その後、制御部 9 は、第 2 の HTML 文書であることを特定するための他の乱数値を生成し、第 1 の処理ステップにおいて生成したセッション値とともに暗号化して第 2 の HTML 文書に付加する。

制御部 9 は、第 2 の HTML 文書を、送受信部 8 を介してクライアント 7 に送信し、さらに、第 2 の HTML 文書の送信と同時に、セッション値、乱数値、処理ステップ情報、第 2 の HTML 文書の入力フォーム

に使用されたクラス定義プログラム毎の設定項目情報を記憶部 10 に記憶（退避）して、第 1 の処理ステップにおいて説明した処理と同様の処理を繰り返す。

第 1 の処理ステップにおいて返信された返信情報に、暗号化されたセッション値および乱数値が含まれている場合であって、暗号化されたセッション値と記憶部 10 に記憶されているセッション値は同一であるが乱数値が異なる場合には、第 1 の HTML 文書に対してクライアント 7 が返信情報を返信したものではないと判断できるので、記憶部 10 に記憶される処理ステップ情報により特定される処理を再度繰り返して実行する。再度処理を繰り返して実行する場合には、制御部 9 は、受信された返信情報を破棄し、再度第 1 の HTML 文書を作成して前述のセッション値と新たに作成した第 1 の HTML 文書を特定するための新たな乱数値とを暗号化して第 1 の HTML 文書に付加する。その後、制御部 9 は送受信部 10 を介して第 1 の HTML 文書を再度クライアント 7 に送信する。再度第 1 の HTML 文書を送付する場合に、制御部 9 は、記憶部 10 に記憶されている乱数値のみを変更して再度記憶部 10 にセッション値、処理ステップ情報等の各種情報を記憶（退避）する。

制御部 9 は、上記の処理を繰り返して、ストリームプログラムに記憶されている処理ステップを順次実行し、一連の処理の終わりに記憶部 10 のデータ記憶部に更新記憶された入力データを、一連の窓口処理における確定したデータとしてデータベース 11 に登録する処理を行った後に、窓口処理が終了した旨の HTML 文書を作成してクライアント 7 に送信して処理を終了する。

次に、簡単なストリームプログラム及びクラス定義プログラムを用いて、より具体的にサーバ 1 とクライアント 7 との間のデータの送受信方法を説明する。

図4～図7は牡蛎料理の注文処理における実行プログラムを示したプログラムリストであり、図4はストリームプログラム、図5、6は、牡蛎料理の注文に関するクラス定義プログラム、図7は、図4に示したストリームプログラムに従って作成されるHTML文書のページ定義を示したプログラムである。

牡蛎料理の注文処理は、基本的に5つの処理ステップによって完了する。ステップ1は、使用されるクラスの宣言及び生成されたインスタンスで用いられる設定項目の初期値設定を行う処理ステップである。

ステップ2は、クライアントに牡蛎の銘柄、調理方法、添える調味料の3種類の設定項目のデータの入力と、コメントの入力を行ってもらう処理ステップであり、この処理を行うために注文入力画面（以下、ページA1とする。）のHTML文書がサーバ1からクライアント7に送信される。図8及び図9はページA1に該当する画面表示を示したものである。

ステップ3は、クライアント7において入力された牡蛎の銘柄、調理方法、添える調味料、コメントの内容を確認してもらう処理ステップである。この処理を行うために、ステップ1においてクライアント7によって入力された注文内容を確認表示させる注文確認画面（以下、ページB1とする。）のHTML文書がサーバ1からクライアント7に送信される。図10及び図11がページB1に該当する画面表示を示したものである。なお、ステップ3では、クライアント7により入力された調味料の種類が2種類以下であるか否かをチェックするエラー処理が実行される。3種類以上の場合には、図12に示すようなエラーメッセージが画面上に表示される。

ステップ4は、注文の確認が為された設定値情報をデータベース11に記憶する処理である。

ステップ5は、ステップ3で確認された内容により牡蛎料理の注文を承った旨を知らせる処理ステップであり、この処理を行うためにステップ3で確認された注文内容と注文処理が完了した旨とを表示する注文完了画面（以下、ページC1とする）のHTML文書がサーバ1からクライアント7に送信される。図13がページC1に該当する画面表示である。

また、図7（a）に示したプログラムは、ページA1のHTML文書を作成するためのページ定義を示したものであり、（b）に示したプログラムは、ページB1のHTML文書を作成するためのページ定義を示したものであり、（c）に示したプログラムは、ページC1のHTML文書を作成するためのページ定義を示したものである。図7（a）～

（c）に示したプログラムは、HTML（Hypertext Markup Language）で記載されており、<h3>～</h3>はHTML文書の表示画面上に見出しを表示させるためのタグである。また、<input type=submit action="next" label="次へ">は、「次へ」と書かれたボタンを表示させるタグであり、クライアント7でこのボタンがクリックされた場合には、返信情報として"next"という戻り値がサーバ1に返信される。<#slot>は、ストリームプログラムの所定の処理ステップにおいて呼び出されたクラス定義プログラムの処理命令に応じてその表示内容が変化することを示したものである。

図5、6に示したクラス定義プログラムを説明する。このクラス定義プログラムは、牡蛎料理の注文に用いられる設定項目ごとのメンバ変数の宣言と、その設定項目情報を用いて処理される処理命令とをまとめたものであり、牡蛎料理の注文に関する処理の際に必要な情報が集約されている。図5、6に示したクラス定義プログラムにおいては、package OysterOrder;に示されるようにOysterOrderというクラスが定義さ

れている。

クラス定義プログラムは、`#BEGIN#DECLARE～#END#DECLARE` で挟まれた部分で示された設定項目のデータ宣言部 p 1 と、`sub ～ { ～ ～ }` で示された処理命令部 p 2 ～ p 7 とで構成されている。

データ宣言部 p 1 では、データベース 1 1 に対応する設定項目毎のメンバ変数の宣言を行っており、注文 ID、牡蛎の銘柄を示すコード、調理方法を示すコード、調味料コードを入力するための符号付整数型のメンバ変数宣言と、調理方法についてのコメントを入力するためのテキスト入力用のメンバ変数宣言とが行われている。クラス定義プログラムがコンパイルされると、コンパイルの度に宣言されたメンバ変数に対応する設定項目がデータベースに作成される。このため、クラス定義プログラムにおいてメンバ変数の設定項目が変更された場合には、クラス定義プログラムのコンパイルの際に自動的にデータベース 1 1 の設定項目も変更される。

処理命令部 p 2 は、クラスで用いられる設定項目毎の記憶領域を、記憶部 1 0 のデータ記憶部に確保することによりインスタンスを生成する処理命令である。

処理命令部 p 3 は、データ記憶部に記憶された入力データを、データベース 1 1 に登録する際に用いるシーケンス番号を生成する処理命令であり、この処理命令を実行させることによって、データベース 1 1 にデータが登録されるときに既に登録されているデータ上に上書き保存されることを防止することが可能となる。

処理命令部 p 4 は、インスタンスで使用される設定項目の初期化を行う処理命令である。この処理命令がストリームプログラムにおいて引用されて実行されると、牡蛎の銘柄の銘柄コードを入力するためのインスタンスのメンバ変数 {brand} に岩牡蛎を示す設定値 1 が代入され、調

理方法コードを入力するためのメンバ変数 {recipe} に生食を示す設定値 1 が代入され、調味料コードを入力するためのメンバ変数

{trimming} にレモンのみを示す設定値 2 が代入され、調理方法のコメントを入力するメンバ変数 {comment} が空欄に設定される。

処理命令部 p 5 は、ページ A 1 に設けられる注文入力フォームを作成するための処理命令である。この処理命令がストリームプログラムで引用されて実行されると、図 8 に示すような HTML 文書（ページ A 1）が作成される。c g i o s 3 の指示に従って制御部 9 は、作成されたページ A 1 に従ってクライアント 7 がデータを入力することが可能な設定項目を、ページ A 1 を作成する際に引用されたインスタンスに関連づけて記憶部 10 に記憶する。クライアント 7 が、ページ A 1 に基づいて牡蛎の銘柄の種類等の入力データ 20 の設定を行ってから、画面上の「次へ」ボタン 21 をクリックすると、サーバ 1 に対して返信情報が送信される。返信情報が送信されたサーバ 1 では、返信情報に含まれる入力データの設定項目が記憶部 10 に記憶される設定項目に該当するか否かの判断を制御部 9 が行い、該当する設定項目がある場合には、記憶部 10 のデータ記憶部に記憶されている設定項目毎の入力データを返信情報に含まれる入力データへ自動的に更新し、該当しない場合には返信情報に含まれる入力データを破棄する。

処理命令部 p 6 は、返信情報に調味料コードに関連する設定項目の入力データが含まれており、さらに記憶部 10 にも調味料コードに関連する設定項目が記憶されている場合であって、互いの設定項目は一致するが、入力コードのデータ値が許容データ値でない場合、例えば、調味料コードが同時に 3 種類選択された場合に、ページ A 1 上にエラーメッセージを表示させる処理命令である。

処理命令部 p 7 は、ページ B 1 において表示される注文確認項目と、

ページ C 1 において表示される注文完了項目とを作成する処理命令である。ページ B 1 とページ C 1 とは画面最上部に表示される見出しの表示と画面の最下部に表示されるボタンの種類が異なるだけで、牡蛎の銘柄、調理方法、調味料、コメントからなる表示項目は同じであるため、ステップ 2 における注文確認画面及びステップ 4 における注文完了画面の両方の HTML 文書を作成する際に用いられる。

次にストリームプログラムについて説明する。ストリームプログラムは、クラス定義プログラムに記載された処理命令を引用することによって一連の窓口処理の流れが記述されたプログラムである。このストリームプログラムを用いることによって、ウェブアプリケーションにおける実行処理順序を「一単位の実行可能なプログラムコード」として記録することが可能となる。

図 4 に示すストリームプログラムは、top、first、second、third、fourth の 5 つの処理ステップによって構成されており、この処理ステップの情報が、処理ステップ情報として記憶部 10 に記憶される。

まず `ustream input_order($o) [# 注文入力ストリーム ~ ~]` において、このストリームプログラムが注文入力ストリームであることを説明するとともに、この注文入力ストリームにおいて使用されるクラスが \$o で示される一つのインスタンスだけであることを宣言している。

```
top:$o = new <APP>::OysterOrder($S);
```

より、new 処理命令（処理命令部 p 2）が実行されることによって OysterOrder で示されるクラス用の記憶領域が確保され、その記憶領域に \$o で示されるインスタンスが生成される。さらに、\$o->default; によって処理命令 default（処理命令部 p 4）が実行されて、\$o で示されるインスタンスの設定項目の初期値設定が行われる。

なお、このストリームプログラムにおいては、1つのインスタンス \$o しか使用されていないが、複数のインスタンスを用いて一連の処理を行う場合には、

```
stream input_order($o1,$o2) [ # 注文入力ストリーム
```

```
top:
```

```
    $o1 = new <APP>::OysterOrder($S);
```

```
    $o1->default;
```

```
    $o2 = new <APP>::OysterOrder($S);
```

```
    $o2->default;
```

に示すように同一のクラスを用いて複数のインスタンス \$o1、\$o2 を宣言することも可能である。

次の処理ステップ first では、図 7 (a) に示したページ定義を利用して、図 8 に示すページ A 1 が作成される。ここで、\$V->[SLOT] = [slot => \$o->input]; と記載されていることから、図 7 (a) の < #slot > のタグの部分に、インスタンス \$o の input 処理命令 (処理命令部 p 5) によって作成される注文入力フォームが、はめ込まれるようにして表示される。処理ステップ first のプログラムには記載されていないが、first の処理ステップにより作成されたページ A 1 には、セッション値と乱数とが暗号化されて付加され、さらに、記憶部 10 にセッション値、乱数値、処理ステップ情報 (ここでは処理ステップが first である旨の情報) が記憶されるとともに、input 処理命令において使用される設定項目情報、具体的には、牡蛎の銘柄、牡蛎の料理方法、添える調味料、コメントの 4 つの設定項目情報がインスタンス \$o に関連付けて記憶される。また、

```
if ($V->[action] eq "next") [ goto second; ]
```

```
goto first;
```

より、ページA 1において図8に示される「次へ」ボタン21がクリックされると、ストリームプログラムにおける次の処理ステップ second に制御部9の処理が移行する。

なお、ページA 1の「次へ」ボタン21がクリックされると処理ステップ second に制御部9の処理が移行することとなるが、処理が移行される前に、既に説明したように、制御部9がクライアント7からの返信情報を取得して、セッション値、乱数値、入力データにおける設定項目情報の抽出・比較処理を行う。制御部9は、返信情報よりセッション値及び乱数値が検出されない場合及びセッション値が一致しない場合には、返信情報を破棄する。また、検出されたセッション値と記憶部10に記憶されたセッション値は同一であるが、検出される乱数値と記憶部10に記憶される乱数値が異なる場合には、クライアント7においてブラウザの「戻る」ボタン22等を押してキャッシュに記憶されている古いHTML文書を表示させた後に、その古いHTML文書に基づいて返信情報が返信されたものと判断することができる。古いHTML文書に基づいて返信情報が返信されたものと判断された場合、制御部9は再度処理ステップ first の処理を実行し、検出されたセッション値及び乱数値が記憶部10に記憶されるセッション値及び乱数値と一致する場合には、返信情報よりクライアント7が入力した入力データの設定項目を検出して、データ記憶部の設定項目に、該当する入力データを更新記憶させる。該当しない項目の入力データに関してはそのデータを破棄してから、次の処理ステップ second に処理を移行する。

次の処理ステップ second では、ページBを作成する前に、制御部9返信情報の内容についてエラーがないかチェックする。

```
if ($o->[ERROR] = $o->check) [ goto first; ]
```

によって、インスタンス\$oのcheck処理命令（処理命令部分p6）の

処理が実行され、クライアント 7 おいて調味料が 3 種類選択された場合には、処理ステップ first に戻って、図 1 2 に示すようにエラー表示を表示させる。

処理ステップ second において、調味料が図 8 に示すように 1 種類しか選択されていない場合には、図 7 (b) に示したページ定義を利用して、図 1 0 に示したページ B 1 が作成される。また、図 9 に示すように、調味料の選択が 2 種類以下である場合には、図 1 1 に示したページ B 1 が作成される。

このページ B 1 の作成の際には、インスタンス \$o の print 処理命令 (処理命令部 p 7) の処理が実行され、処理ステップ first において返信情報としてクライアント 7 からサーバ 1 に返信されたページ A 1 の入力データが表示される。ページ B 1 においては、ページ A 1 と違って「注文実行」ボタンと、「戻る」ボタンとが表示される。ここで「注文実行」ボタンがクリックされると次の処理ステップである third に処理が進み、「戻る」ボタンをクリックすると一つ前の処理ステップである first の処理に戻る事となる。

次の処理ステップ third では、インスタンス \$o に用いられるメンバ変数 {orid} に対して nextorid 処理命令 (処理命令部 p 3) を実行することによってシーケンス番号を代入し、\$\$S[DA]-

>saveObject(\$o);により、シーケンス番号が入力されたメンバ変数 {orid} を主キーとしてインスタンス \$o の設定項目毎の入力データがデータベース 1 1 に記憶される。

その後、処理ステップが fourth に進んで、図 7 (c) に示したページ定義を利用して、図 1 3 に示したページ C 1 が作成される。処理ステップ fourth により一通りの牡蛎料理の注文処理が終了するが、ビュー C 1 に表示される「最初に戻る」ボタンをクリックすることによって、

再度処理ステップ first の処理を繰り返し行うことが可能である。

以上説明したように本発明に係るサーバ 1 は、サーバ 1 が実行する一連の窓口処理をサーバ 1 でその処理ステップの処理状況を確認しながら実行することが可能となる。このため、一連の窓口処理が手順に従って行われるか否かがクライアント 7 に委ねられていた従来のウェブアプリケーションサーバに較べて、本発明に係るサーバ 1 は、一連の窓口処理が処理手順に従って実行されるように、サーバ 7 に送信する HTML 文書の表示内容を制御することができ、サーバ 1 側の裁量で窓口処理を進めることによって意図しない処理の遷移を防止することが可能となる。

また、実行プログラムを作成するプログラマーはストリームプログラムを作成するだけで、クライアントの特定及び処理ステップ毎の返信情報の確認等の処理をプログラム上に記述しなくても、クライアントの特定、返信情報における入力データの保存等が確実に実行されるので、プログラム作成の労力が低減されるとともに、プログラムの修正・変更処理が容易になる。

さらに、本発明に係るサーバ 1 は、cgi os 3 の指示により、返信情報に含まれる入力データの設定項目の種類が、送信した HTML 文書の入力フォームにおいて入力可能な設定項目の種類に対応するものであるか否かを制御部 9 が判断して、入力データの設定項目と記憶されたが設定項目とが異なる場合にはその入力データを破棄し、一致する場合にはその入力データをインスタンスに関連付けて自動的に記憶部 10 のデータ記憶部に記憶させることができる。このため、本発明に係るサーバ 1 において、意図しない入力データがクライアント 7 からの返信情報として記憶されてその後の処理に反映されることを防止することができる。

また、プログラマーは、返信情報に含まれる入力データが適正なデータであるか否かの判断を行う処理をプログラムで規定しておく必要がな

いので、プログラム作成労力の一層の低減を図ることが可能となるとともに、サーバ 1 の安定性向上を図ることが容易となる。

さらに、HTML 文書の送信時及び返信情報の受信時にのみ、記憶部 10 に記憶された必要なデータを読み出して（復帰させて）制御部 9 に必要な処理を行わせることができるので、制御部 9 は複数のストリームプログラムを用いたマルチタスク処理を容易に実現することが可能となる。

また、クラス定義プログラムごとに一連の窓口処理で使用する設定項目毎のメンバ変数とその設定項目を利用する処理命令とがまとめられており、他のクラス定義プログラムやストリームプログラムにおいてメンバ変数は記載されないので、一連の窓口処理で使用するメンバ変数が変更される場合であっても、クラス定義プログラム内のメンバ変数の宣言内容を変更するだけで良い。このため、他のクラス定義プログラム及びストリームプログラム等を変更することなく、クラス定義プログラムに記載されるメンバ変数の設定項目や処理命令を変更するだけで容易に仕様変更を行うことが可能となり柔軟性に富んだシステム構築を行うことが可能となる。特に、c g i o s 3 では、同一ストリームプログラムにおいて同一のクラスから生成された複数のインスタンスが用いられている場合であっても、c g i o s 3 がクラスのメンバ変数を、各インスタンスの設定項目毎に管理するので、プログラマーはストリームにおいて使用されるメンバ変数の設定項目が重複する場合であっても、プログラム上でその違いを明確にすることなくストリームプログラムを作成することができる。

例えば、図 14 に示すように、1 画面において牡蛎の銘柄、牡蛎の調理方法、添える調味料、コメントの項目を 2 組表示させるページ D を作成する場合、ストリームプログラムにおいては、\$o1 と \$o2 との 2 つ

のインスタンスを宣言することによって、OysterOrder クラスに使用されるメンバ変数の設定項目毎のデータを、\$o1 と \$o2 とからなる識別子を用いてそれぞれを区別し、別々に記憶する。ビューDをクライアント7に送信する場合には、入力フォームにおけるインスタンス\$o1 の設定項目（牡蛎の銘柄、牡蛎の調理方法、添える調味料、コメント）とインスタンス\$o2 の設定項目（牡蛎の銘柄、牡蛎の調理方法、添える調味料、コメント）とを区別して記憶部10に記憶させ、返信情報より入力データを検出する場合も、インスタンス\$o1 の入力データの設定項目（牡蛎の銘柄、牡蛎の調理方法、添える調味料、コメント）とインスタンス\$o2 の入力データの設定項目（牡蛎の銘柄、牡蛎の調理方法、添える調味料、コメント）とを区別して抽出し、それぞれデータ記憶部に別々の入力データとして記憶する。

具体的に、図19を用いて、図14に示す入力フォームの設定項目のうち“牡蛎の銘柄”欄と“コメント”欄だけに注目して、上述した処理を説明する。図19に示すように、サーバ1の制御部9は、cgios3を用いて“牡蛎の銘柄”、“コメント”の設定項目を2つの識別子\$o1 と \$o2 とを用いてそれぞれ区別し、一意に区別される通し番号（例えば、“a”、“b”、“c”・・・）を割り付けて記憶部10に記憶させる。制御部9は、インスタンス\$o1 の設定項目として、牡蛎の銘柄に関するデータであって1～3の数値により選択される選択形式のデータ“a”と、コメントに関するデータであってテキスト形式のデータであるデータ“b”とを記憶部10に入力フォーム台帳として記憶させる。同様に制御部9は、インスタンス\$o2 の設定項目として、牡蛎の銘柄に関するデータであって1～3の数値により選択される選択形式のデータ“c”と、コメントに関するデータであってテキスト形式の

データであるデータ“e”とを記憶部10に入力フォーム台帳として記憶させる。

その後、制御部9は、2つの入力フォーム20を含むHTML文書を作成してクライアント7に送信する。クライアント7により入力フォームに対応する返信情報が返信されると、制御部9は、返信情報の設定項目が記憶部10に記憶された入力フォーム台帳の設定項目に該当するかどうかの判断を行う。該当する場合、制御部9は、返信情報に含まれるデータ“a”がインスタンス\$ o 1の“牡蛎の銘柄”に関するデータであり、データ“b”がインスタンス\$ o 1の“コメント”に関するデータであり、データ“c”が、インスタンス\$ o 2の“牡蛎の銘柄”に関するデータであり、データ“d”がインスタンス\$ o 2の“コメント”に関するデータであると判断し、それぞれのデータをデータ記憶部に記憶する。

このように、重複する設定項目からなる入力フォームをHTML文書で作成する場合であっても、各設定項目に対応する処理命令を1つ用意して、同一クラスを用いて複数のインスタンスを生成することによって、各設定項目の区分けをc g i o sで自動的行わせることができ、クラス定義プログラムの処理命令だけを変更するだけで容易に入力フォームの表示形式、設定項目等の仕様変更を容易に行うことができる。さらに、クラス定義プログラムで使用するメンバ変数は他のクラス定義プログラムやストリームプログラム等に依存しないので、クラス定義プログラムの設定項目等を変更しても、その変更に起因するシステム全体のエラーの発生を低減させることができる。

なお、c g i o s 3は、一連の窓口処理の流れを示すストリームプログラムと、このストリームプログラムで使用するインスタンスのメンバ変数とそのメンバ変数を用いる処理命令とでまとめたクラス定義プロ

グラムとを組み合わせることでプログラム構成を行うので、今までのHTML文書の作成技法では困難であったと入力フォームを容易に作成することが可能となる。例えば、注文されたデータの一覧表をHTML文書において作成する場合には、表外枠、見出し、各種設定項目等をインスタンスの処理命令により特定の出力形式で作成し、作成された一覧表をストリームプログラムにおいて引用することによって、一覧表をHTML文書に挿入するようにして貼り付けることが可能となるので、各行が入力フォームを持つような表一覧を簡単に作成することができる。さらに、ストリームプログラムから他のストリームプログラムを呼び出すことも可能であるため、上述の牡蛎料理の注文処理を示したストリームプログラムにおいて、表一覧を作成する表一覧作成ストリームを呼び出し、この表一覧作成ストリームにおいてインスタンスから表一覧を作成するための処理命令を引用して表一覧を作成し、表一覧作成後に牡蛎調理の注文処理のストリームに処理を戻すことも可能である。さらに、表一覧を作成する方法を応用することによって、HTML文書における表一覧の他に、Microsoft excel（登録商標）に対して一覧表形式にデータを出力することも可能である。

また、ストリームプログラムにおいて他のストリームプログラムを呼び出す方法とHTML文書を作成するためのページ定義とを活用することによってウェブページにおけるウィンドウ管理をサーバ1で行うことが可能となる。従来のウェブアプリケーションでは、基本的に1つのウェブページでは1つの処理しか実行することができず、複数の処理を行う場合にはクライアント7が使用するブラウザのフレーム機能を用いることが一般的であった。しかしながらブラウザのフレーム機能の動作は、基本的にクライアント7の裁量に委ねられているので、サーバ1が意図する形式のウィンドウ表示とならない場合があった。cgios3では、

HTML文書のページ定義によりウィンドウ表示形式を決定し、各ウィンドウ内で個別にストリームプログラムを実行させて、各ストリームプログラムの実行結果として出力されるHTML文書を該当するウィンドウに貼り付けて1つのウェブページを生成することができるので、ブラウザのフレーム機能に依存することなくウィンドウ管理を行うことが可能となる。

さらに、ウィンドウ毎に実行される各ストリームプログラムの処理に応じて記憶部10に記憶されたデータの読み出し（復帰）、記憶（退避）を行うことができるので、マルチウィンドウを用いたマルチタスク処理を簡易に実現することが可能となる。

図15に示すような3つのウィンドウを有するHTML文書をc g i o s 3で作成する方法を、図16に示したストリームプログラムと、図17に示したページ定義とを用いて説明する。まず、図16(a)に示した起動ストリームプログラムにより、ウィンドウの設定がされていないウェブページ（起動ストリームプログラムのp_fork

("", "/ / ", "this") の" this "がその旨を示している) 上で図16(b)に示すウィンドウ起動ストリームプログラムを実行し、p_wait () により起動ストリームプログラムの処理の待機を行う。

起動ストリームプログラムにおいてウィンドウ起動ストリームプログラムが実行されると、図17(a)に示したページ定義が用いられた図15に示すような3つのウィンドウが形成され、このウィンドウの"menu"部分において図16(c)に示すメニューストリームプログラムが実行される。

メニューストリームプログラムが実行されて、ウィンドウの"menu"部分に図17(b)に示したページ定義のメニュー表示が表示され、メニュー表示において注文入力をクリックされるとウィンドウの

view 部分において牡蛎料理の注文処理に関するストリームプログラムが実行されて、図 18 に示すように牡蛎料理の注文に関する入力フォームが作成される。また、メニュー表示において注文一覧表示がクリックされると、注文一覧表示を行うストリームプログラムが実行される。

このように、複数のページ定義と、複数のストリームプログラムを活用することによって、一連の窓口処理の内容に応じて入力フォーム等を表示させるウィンドウを使い分けることが可能となる。さらに、クライアント 7 のブラウザにおけるフレーム機能に依存されずにウィンドウ管理を行うことができるので、サーバ 1 側で意図したウィンドウ表示をクライアント 7 のウェブページ上で表示させることが可能となる。

以上本発明に係るサーバを説明してきたが、本発明に係るサーバは上記発明の実施の形態に示したものに限定されるものではない。例えば、本発明の実施の形態においては、`c g i o s 3` が OS 上で動くミドルウェアであるものとして説明を行ったが、ウェブサーバ用アプリケーション（例えば Apache 等）において実行される支援ソフトウェア（支援プログラム）であっても良い。さらに、`c g i o s 3` 自体がオペレーションシステムであって、オペレーションシステムにおいて `c g i o s` 特有の機能を実現させたものであっても良い。

また、サーバ 1 が `c g i o s 3` の機能を利用して処理を行う場合に、必ずしもストリームプログラムを用いる必要はなく、ストリームプログラムを用いずに、クライアント 7 に送信する入力フォームの設定項目を入力フォーム台帳として記憶部 10 に記憶し、クライアント 7 から返信された返信情報の設定項目と記憶部 10 に記憶された設定項目とを対比して受信した設定項目の種類が同一であるか否かの判断を行うようにしてもよい。

例えば、制御部 9 は、図 20 に示すように、入力フォームにおいてクライアント 7 が入力可能な設定項目に対してそれぞれ通し番号 “a” ～ “c” を割り付けて記憶部 10 に入力フォーム台帳として記憶させる。このとき制御部 9 は、牡蛎の銘柄に関するデータであって 1 ～ 3 の数値により選択される選択形式のデータをデータ “a” とし、コメントに関するデータであってテキスト形式のデータをデータ “b” とし、クライアント 7 に送信する HTML 文書に設定されるボタンを押下したことを示すボタンアクションデータをデータ “c” として記憶部 10 に記憶させる。

その後、制御部 9 は、クライアント 7 から返信された返信情報を受信して返信情報からデータ “a” ～ “c” に該当する情報を抽出し、記憶部 10 に入力フォーム台帳として記憶される設定項目の種類と返信情報の “a” ～ “c” の設定項目の種類とを比較して一致する場合には、牡蛎の銘柄、コメントのデータをデータ記憶部に記憶し、一致しない場合には受信した返信情報を破棄する。例えば、返信情報に含まれるデータ “a” の値が 9 である場合、制御部 9 は、入力フォーム台帳よりデータ “a” が 1 ～ 3 の数値により選択される選択形式のデータであることを判断し、データ “a” が 9 である返信情報が不適切なものであるものと判断して返信情報を破棄する。

このように、ストリームプログラムの有無に拘わらずサーバ 1 がクライアント 7 に送信する HTML 文書の入力フォームの設定項目を記憶部 10 に記憶させ、クライアント 7 より受信した返信情報の設定項目が記憶部 10 に記憶された入力データの設定項目と同一種類か否かを判断することによって、意図しない入力データが返信情報としてデータ記憶部に記憶されてその後の処理に反映されることを防止することが可能となる。

また、サーバ1がc g i o s 3の機能を利用して、記憶部10に記憶させる入力フォームの情報を、サーバ1の処理に適したデータ形式からクライアント7でのデータ入力に適したデータ形式に加工・変更を行った後にクライアント7に送信し、返信された返信情報を加工・変更を行う前の形式に復元した後にサーバ1において処理を実行する様にしてもよい。

例えば、UNIXは、1970年1月1日0時0分0秒からの積算秒数を用いて一般的な時間の管理を行っている。このため、入力フォームにおいてクライアント7に“年”、“月”、“日”、“分”を別々に分けて選択させる場合、c g i o s 3では、積算秒数で示される日時情報（以下、積算情報とする）と、“年”、“月”、“日”、“分”等の単位で表示される日時情報（以下、時分情報とする）との変換を行う”C a l e n d a r タグ”処理モジュールを利用して、積算情報と時分情報との変換を行ってもよい。具体的には、図21に示すように、“年”に関するデータであって西暦表記された2000、2001・・・の数値により選択される選択形式のデータをデータ“a”とし、“月”に関するデータであって1～12の数値により選択される選択形式のデータをデータ“b”とし、“日”に関するデータであって1～31の数値により選択される選択形式のデータをデータ“c”として記憶部10に記憶させ、クライアント7に、入力フォームとして日時情報が加えられてHTML文書を送信する。

クライアント7から返信情報が返信されると、サーバ1は受信した返信情報よりデータ“a”、“b”、“c”の設定項目が記憶部10に記憶された設定項目と同一種類のものかどうかの判断を行い、異なる場合には返信情報を破棄し、同一である場合には、c g i o s 3の”C a l

“e n d a r タグ” 処理モジュールを利用して返信情報に含まれる時分情報を積算情報に変換し、データ記憶部に記憶する。

このように、c g i o s 3において、サーバ1での処理に適したデータ形式を入力フォームの設定項目に応じたデータに変換し、変換したデータを入力フォーム台帳として記憶部10に記憶させるとともに、クライアント7から返信された返信情報の設定項目と記憶部10に記憶される設定項目とを比較し、設定項目の種類が一致する場合には、返信情報をサーバ1の処理に適したデータ形式に変換することによって、プログラマが返信情報に含まれる入力データの変換処理プログラムを作成することなく返信情報のデータをサーバ1に適したデータ形式で使用する事が可能となる。

さらに、入力フォームとしてチェックボックスが用意されている場合、入力フォームの設定項目の種類に加えて、チェックボックスをチェック済にした設定項目を特定してその項目がチェック済となっている旨を記憶部10に記憶させてもよい。

具体的には、クライアント7に送信するHTML文書にチェックボックス欄を設け、そのチェックボックス欄を予めチェック済としてHTML文書を作成してクライアント7に送信する。このとき制御部9は、入力フォームの設定項目の他に、その設定項目のうちチェックボックスをチェック済みとした項目にチェックボックスがチェック済として設定されている旨のデータを対応付けて記憶部10に記憶させる。

その後、制御部9は、クライアント7より返信された返信情報の設定項目と入力フォーム台帳の設定項目とを比較し、入力フォーム台帳においてチェック済として記録されたチェックボックスの設定項目に対応する情報が、返信情報に含まれているか否かを判断する。通常、クライアント7から返信されるチェックボックスに関する情報は、チェックボッ

クスがチェックされている場合には返信情報に含まれて返信されるが、チェックボックスがチェックされていない場合（チェック済であったものをチェックなしに変更した場合を含む）には含まれない。このため、制御部 9 は、返信された返信情報の設定項目のうち、チェック済としたチェックボックスに関する情報が含まれているか否かを、入力フォーム台帳の情報を参照しながら判断する。チェック済として記録されたチェックボックスに関する情報が返信情報に含まれていない場合、制御部 9 は、クライアント 7 が意図的にチェックボックスのチェックを外したものと判断する。

このように、入力フォーム台帳としてチェックボックスの設定項目の種類とチェック済とされたチェックボックスの設定項目とを記憶部 10 に記憶させることにより、記憶部 10 にチェック済として記憶されたチェックボックスに関する情報が返信された返信情報に含まれていない場合には、クライアント 7 が意図的にチェックを外したものと判断することができ、チェックボックスにおいてチェックがされていないという情報を積極的に知得することが可能となる。また、これらの判断は c g i o s 3 の機能を用いることによって、チェックボックスのチェックが外されたか否かの判断をプログラムによって実現させる必要性がなくなるので、プログラムの作成労力・作成工程の低減を図ることが可能となる。

請求の範囲

1. 通信装置を用いてクライアントにウェブ情報を送信し、該通信装置において前記ウェブ情報に対する前記クライアントからの返信情報を受信して該返信情報を記憶装置に記憶させ、前記記憶装置に記憶される一連の処理の処理ステップに従って、前記返信情報に基づく処理を実行するウェブアプリケーションサーバの制御部に、

前記クライアントを特定するためのクライアント特定情報と前記ウェブ情報を特定するためのウェブ特定情報とを当該ウェブ情報に付加させる情報付加処理と、

前記クライアント特定情報と前記ウェブ特定情報とが付加された前記ウェブ情報を特定の前記処理ステップにおいて前記通信装置を用いて前記クライアントに送信させる情報送信処理と、

前記ウェブ情報に付加された前記クライアント特定情報及び前記ウェブ特定情報と、前記ウェブ情報が前記クライアントに送信されたときの前記処理ステップに関する処理情報とを、前記ウェブ情報が前記クライアントに送信される度に前記記憶装置に記憶させる情報記憶処理と、

前記クライアントより返信される返信情報を、前記通信装置において受信させる情報受信処理と、

受信された返信情報にクライアント特定情報とウェブ特定情報とが含まれていないかを検出させる返信情報検出処理と、

前記返信情報よりクライアント特定情報及びウェブ特定情報が検出され、かつ、検出された前記クライアント特定情報及び前記ウェブ特定情報が前記記憶装置に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一である場合に、前記記憶装置に記憶された前記処理情報より求められる処理ステップの次の処理ステップを前記記憶装置より求めて実行させ、前記返信情報よりクライアント特定情報及びウェブ特

定情報が検出され、かつ、検出された前記クライアント特定情報と前記記憶装置に記憶される前記クライアント特定情報とは同一であるが、前記返信情報より検出されたウェブ特定情報と前記記憶装置に記憶される前記ウェブ特定情報とが異なる場合には、前記記憶装置に記憶された前記処理情報より求められる処理ステップを再度実行させる業務実行処理と

を実行させることを特徴とするウェブアプリケーションサーバ用支援プログラム。

2. 前記記憶装置は、メンバ変数の設定項目別の作業領域として前記一連の処理において使用されるデータ宣言部と、前記一連の処理において前記返信情報に含まれるデータを前記設定項目毎に記憶するデータ記憶部とを備え、

前記制御部に対して

前記ウェブ情報に前記クライアントにデータを入力させるためのデータ入力フォームが含まれている場合には、当該ウェブ情報を前記クライアントに送信する際に前記データ入力フォームにより入力される入力データの設定項目の種類を前記データ宣言部より判別し、前記情報記憶処理において前記クライアント特定情報、前記ウェブ特定情報、前記処理情報とともに前記記憶装置に記憶させ、

前記返信情報より検出されたクライアント特定情報及びウェブ特定情報が前記記憶装置に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一であって、前記返信情報より検出された入力データの設定項目の種類と前記記憶装置に記憶される入力データの設定項目の種類とが同一である場合には、前記業務実行処理において、検出された前記入力データを設定項目毎に前記データ記憶部に記憶させ

ることを特徴とする請求項 1 に記載のウェブアプリケーションサーバ用支援プログラム。

3. 前記ウェブ情報に前記データ入力フォームが複数含まれている場合であって、前記データ入力フォームに従って入力される入力データの設定項目の種類が重複する場合には、
前記制御部に、

設定項目の種類が重複する入力データの識別を行うための識別子を前記データ入力フォーム毎に付して前記ウェブ情報に含ませる識別子追加処理を実行させるとともに、

前記情報記憶処理において、前記データ入力フォームにおいて入力される入力データの設定項目の種類を前記識別子に関連付けて前記記憶装置に記憶させ、

前記業務実行処理において、前記返信情報より検出される前記入力データの設定項目の種類と前記記憶装置に記憶される設定項目の種類とが同一識別子毎に一致する場合には、検出された前記入力データの設定項目を前記記憶装置に記憶させた識別子に基づいて区別して設定項目毎に前記データ記憶部に記憶させ

ることを特徴とする請求項 2 に記載のウェブアプリケーションサーバ用支援プログラム。

4. 前記情報記憶処理の後に、前記情報受信処理を行うために必要な情報を前記記憶装置に退避させて前記制御部に他の処理を実行させ、

前記情報受信処理により前記返信情報を受信したときに、前記退避された情報を復帰させて前記制御部に前記返信情報検出処理を再開させ

ることで前記ウェブアプリケーションサーバにおいて並列処理を実現させることを特徴とする請求項 1 に記載のウェブアプリケーションサーバ用支援プログラム。

5. クライアントにデータを入力させるためのデータ入力フォームが含まれるウェブ情報を通信装置を用いて前記クライアントに送信し、該通信装置において前記データ入力フォームに従って入力された入力データを返信情報として前記クライアントから受信し、メンバ変数の設定項目別の作業領域として使用されるデータ宣言部と、前記返信情報に含まれるデータを前記設定項目毎に記憶するデータ記憶部とを備える記憶装置に前記入力データを記憶させるウェブアプリケーションサーバの制御部に対して、

当該ウェブ情報に前記クライアントにデータを入力させるためのデータ入力フォームを付加させる情報付加処理と、

前記データ入力フォームが付加された前記ウェブ情報を前記通信装置を用いて前記クライアントに送信させる情報送信処理と、

前記データ入力フォームにより入力される入力データの設定項目の種類を前記記憶装置に記憶される前記データ宣言部より判別して、前記ウェブ情報が前記クライアントに送信される度に前記記憶装置に記憶させる情報記憶処理と、

前記クライアントより返信される返信情報を、前記通信装置において受信させる情報受信処理と、

受信された返信情報に前記入力データが含まれていないかを検出させる返信情報検出処理と、

前記返信情報より前記入力データが検出され、かつ検出された入力データの設定項目の種類と、前記記憶装置に記憶される設定項目の種類とが同一である場合には、検出された前記入力データを設定項目毎に前記データ記憶部に記憶させる業務実行処理と

を実行させることを特徴とするウェブアプリケーションサーバ用支援プログラム。

6. 前記ウェブ情報に前記データ入力フォームが複数含まれている場合であって、前記データ入力フォームに従って入力される入力データの設定項目の種類が重複する場合には、前記制御部に、

設定項目の種類が重複する入力データの識別を行うための識別子を前記データ入力フォーム毎に付して前記ウェブ情報に含ませる識別子追加処理を実行させるとともに、

前記情報記憶処理において、前記データ入力フォームにおいて入力される入力データの設定項目の種類を前記識別子に関連付けて前記記憶装置に記憶させ、

前記業務実行処理において、前記返信情報より検出される前記入力データの設定項目の種類と前記記憶装置に記憶される設定項目の種類とが同一識別子毎に一致する場合には、検出された前記入力データの設定項目を前記記憶装置に記憶させた識別子に基づいて区別して設定項目毎に前記データ記憶部に記憶させ

ることを特徴とする請求項5に記載のウェブアプリケーションサーバ用支援プログラム。

7. 前記情報記憶処理の後に、前記情報受信処理を行うために必要な情報を前記記憶装置に退避させて前記制御部に他の処理を実行させ、

前記情報受信処理により前記返信情報を受信したときに、前記退避された情報を復帰させて前記制御部に前記返信情報検出処理を再開させ

ることで前記ウェブアプリケーションサーバにおいて並列処理を実現させることを特徴とする請求項5に記載のウェブアプリケーションサーバ用支援プログラム。

8. 通信装置を用いてクライアントにウェブ情報を送信し、該通信装置において前記ウェブ情報に対する前記クライアントからの返

信情報を受信して該返信情報を記憶装置に記憶させ、前記記憶装置に記憶される前記一連の処理の処理ステップに従って、前記返信情報に基づく処理を制御部が実行するサーバにおいて、

前記クライアントを特定するためのクライアント特定情報と前記ウェブ情報を特定するためのウェブ特定情報とを当該ウェブ情報に付加させる情報付加処理と、

前記クライアント特定情報と前記ウェブ特定情報とが付加された前記ウェブ情報を特定の前記処理ステップにおいて前記通信装置を用いて前記クライアントに送信する情報送信処理と、

前記ウェブ情報に付加された前記クライアント特定情報及び前記ウェブ特定情報と、前記ウェブ情報が前記クライアントに送信されたときの前記処理ステップに関する処理情報とを、前記ウェブ情報が前記クライアントに送信される度に前記記憶装置に記憶させる情報記憶処理と、

前記クライアントより返信される返信情報を、前記通信装置に受信させる情報受信処理と、

受信された返信情報にクライアント特定情報とウェブ特定情報とが含まれていないかを検出する返信情報検出処理と、

前記返信情報よりクライアント特定情報及びウェブ特定情報が検出され、かつ、検出された前記クライアント特定情報及び前記ウェブ特定情報が前記記憶装置に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一である場合に、前記記憶装置に記憶された前記処理情報より求められる処理ステップの次の処理ステップを前記記憶装置より求めて実行し、前記返信情報よりクライアント特定情報及びウェブ特定情報が検出され、かつ、検出された前記クライアント特定情報と前記記憶装置に記憶される前記クライアント特定情報とは同一であるが、前記返信情報より検出されたウェブ特定情報と前記記憶装置に記憶される前

記ウェブ特定情報とが異なる場合には、前記記憶装置に記憶された前記処理情報より求められる処理ステップを再度実行する業務実行処理とを前記制御部が実行することを特徴とするサーバ。

9. 前記記憶装置は、メンバ変数の設定項目別の作業領域として前記一連の処理において使用されるデータ宣言部と、前記一連の処理において前記返信情報に含まれるデータを前記設定項目毎に記憶するデータ記憶部とを備え、

前記制御部は、

前記ウェブ情報に前記クライアントにデータを入力させるためのデータ入力フォームが含まれている場合に、当該ウェブ情報を前記クライアントに送信する際に前記データ入力フォームにより入力される入力データの設定項目の種類を前記データ宣言部より判別し、前記情報記憶処理において前記クライアント特定情報、前記ウェブ特定情報、前記処理情報とともに前記記憶装置に記憶し、

前記返信情報より検出されたクライアント特定情報及びウェブ特定情報が前記記憶装置に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一であって、前記返信情報より検出された入力データの設定項目の種類と前記記憶装置に記憶される入力データの設定項目の種類とが同一である場合には、前記業務実行処理において、検出された前記入力データを設定項目毎に前記データ記憶部に記憶する

ことを特徴とする請求項8に記載のサーバ。

10. 前記ウェブ情報に前記データ入力フォームが複数含まれている場合であって、前記データ入力フォームに従って入力される入力データの設定項目の種類が重複する場合には、

前記制御部は、

設定項目の種類が重複する入力データの識別を行うための識別子を前

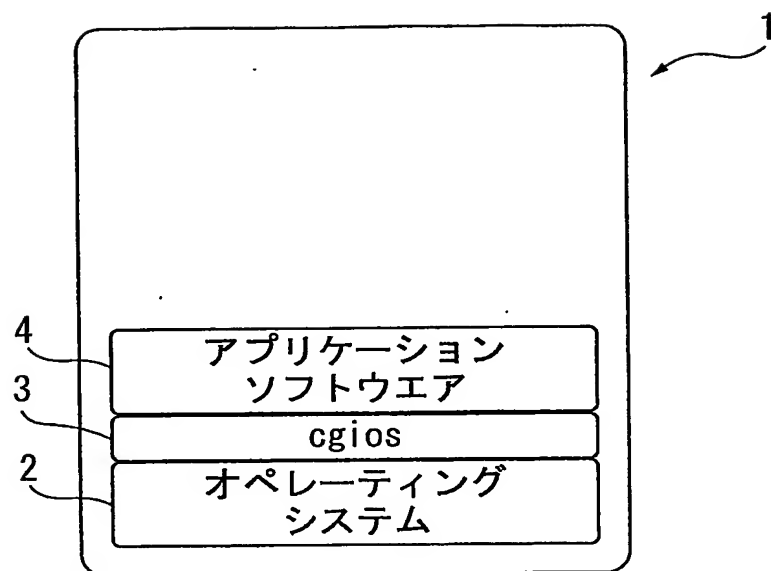
記データ入力フォーム毎に付して前記ウェブ情報に含ませる識別子追加処理を行うとともに、

前記情報記憶処理において、前記クライアント特定情報、前記ウェブ特定情報、前記処理情報とともに、前記データ入力フォームにおいて入力される入力データの設定項目の種類を前記識別子に関連付けて前記記憶装置に記憶させ、

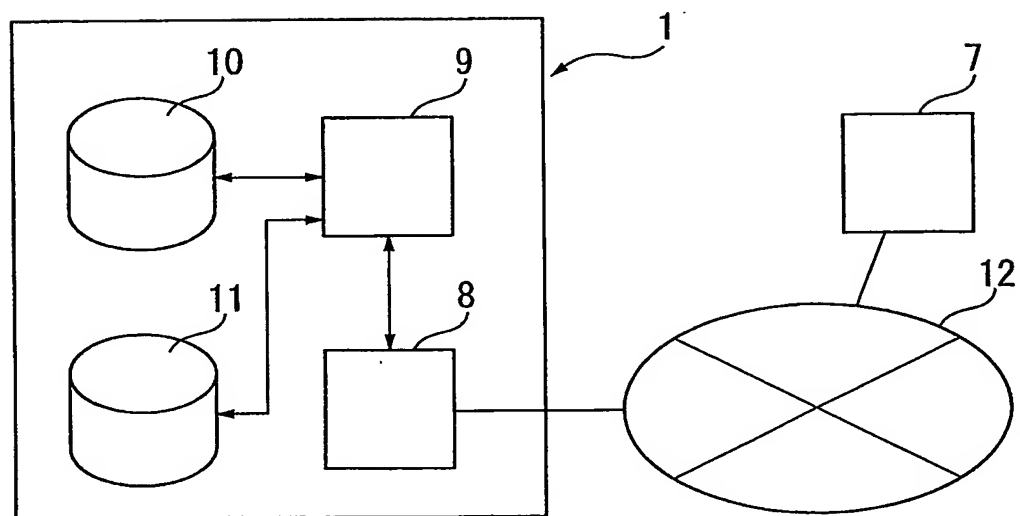
前記業務実行処理において、前記返信情報より検出されたクライアント特定情報及びウェブ特定情報が前記記憶装置に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一であって、前記返信情報より検出される前記入力データの設定項目の種類と前記記憶装置に記憶される設定項目の種類とが同一識別子毎に一致する場合には、検出された前記入力データの設定項目を前記記憶装置に記憶させた識別子に基づいて区別して設定項目毎に前記データ記憶部に記憶させ

ることを特徴とする請求項 8 に記載のサーバ。

第1図

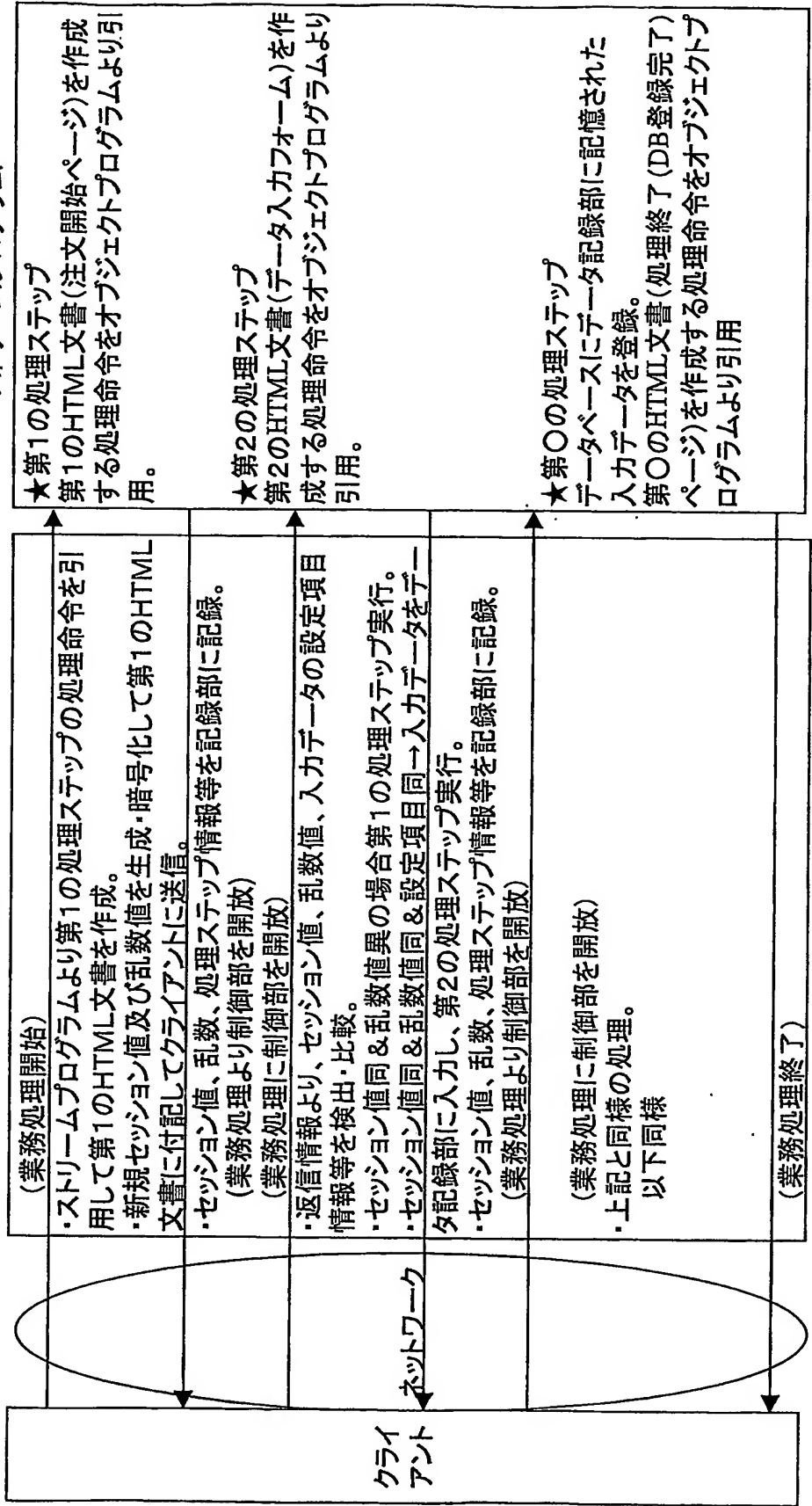


第2図



第3図

cgios



第4図

```
ustream input_order($o) { # 注文入力ストリーム
  top:
    $o = new <APP>::OysterOrder($S);
    $o->default;

  first:
    page(input_order.page) {
      $V->{SLOT} = { slot => $o->input };
    }
    if ($V->{action} eq "next") { goto second; }
    goto first;

  second:
    if ($o->{ERROR} = $o->check) { goto first; }
    page(print_order_conf.page) {
      $V->{SLOT} = { slot => $o->print };
    }
    if ($V->{action} eq "back") { goto first; }
    elsif ($V->{action} eq "next") { goto third; }
    goto second;

  third:
    $o->{orid} = $o->nextorid;
    $$S{DA}->saveObject($o);

  fourth:
    page(print_order_done.page) {
      $V->{SLOT} = { slot => $o->print };
    }
    if ($V->{action} eq "next") { goto first; }
    goto fourth;
}
```

第5図

牡蛎料理の注文を表現するオブジェクト。メンバ変数の詳細は <MEMBER> である。

```
package OysterOrder;
```

```
# 牡蛎料理の注文を表現するオブジェクト
```

```
#BEGIN_DECLARE
```

```
$data = "orid int,                # 注文ID
        brand int,               # 牡蛎の銘柄コード
        recipe int,             # 調理方法コード
        trimming int,           # 調味料コード
        comment text,           # 調理方法についてのコメント
        primary key(orid)";
```

p 1

```
$seqs = "seq_orid:orid:1,";
```

```
$defo = "brand => OysterType, recipe => RecipeType, trimming => TrimmingType";
```

```
#END_DECLARE
```

```
use strict;
```

```
sub new { # 牡蛎料理注文オブジェクトのコンストラクタ
    my($type,$$,%hash) = @_ ;
    my %h = ("S" => $$, "SELF" => "OysterOrder", %hash);
    return bless \%h, "<APP>::OysterOrder";
}
```

p 2

```
sub nextorid { # 次の ORID
    my($V) = @_ ; my $$ = $V->{S};
    my($orid) = $$S{DA}->getrow("select nextval('seq_orid')");
    $orid;
}
```

p 3

```
sub default { # デフォルト値をセット
    my($V) = @_ ; my $$ = $V->{S};
    $V->{brand} = 1; # 岩牡蛎
    $V->{recipe} = 1; # 生食
    $V->{trimming} = 2; # レモンのみ
    $V->{comment} = "";
    delete $V->{orid};
}
```

p 4

第6図

```

sub input { # 注文入力フォーム
    my($V) = @_ ; my $S = $V->{S};
    print_webform( # 注文入力フォーム
'<table>
    <ifdef ERROR><span class=error><!ERROR></span></ifdef>
    <tr><td>牡蛎の銘柄</td><td>
        <input type=select name=brand value=$brand choice=[OysterType]></td></tr>
    <tr><td>牡蛎の調理方法</td><td>
        <input type=select name=recipe value=$recipe choice=[RecipeType]></td></tr>
    <tr><td>添える調味料</td><td>
        <input type=TrimmingType name=trimming value=$trimming></td></tr>
    <tr><td>コメント</td><td>
        <input type=text name=comment value=$comment size=60></td></tr>
</table>');
}

```

p 5

```

sub check {
    my($V) = @_ ; my $S = $V->{S};
    my $count = 0;
    my $value = $V->{trimming};
    while ($value>0) { $count += $value & 1; $value >>= 1; }
    my $error = "";
    if ($count >= 3) {
        $error = "<i>同時に添付できる調味料は2種類までです。";
    }
    $error;
}

```

p 6

```

sub print { # 注文の HTML 形式での清書
    my($V) = @_ ; my $S = $V->{S};
    print_webtext( # 注文の HTML 形式での清書
'<table>
    <tr><td>牡蛎の銘柄</td><td><%brand></td></tr>
    <tr><td>牡蛎の調理方法</td><td><%recipe></td></tr>
    <tr><td>添える調味料</td><td><%trimming></td></tr>
    <tr><td>コメント</td><td><%comment></td></tr>
</table>');
}

```

p 7

第7図

(a) <注文を入力するページ:/main/input_order.page>

```
<h3>牡蛎の銘柄・調理方法を選択してください</h3>
<#slot>
<br>
<input type=submit action="next" label="次へ">
```

(b) <注文を確認するページ:/main/print_order_conf.page>

```
<h3>御注文を確認ください。</h3>
<#slot>
<br>
<input type=submit action="next" label="注文実行">
<input type=submit action="back" label="戻る">
```

(c) <注文の完了を通知するページ:/main/print_order_done.page>

```
<h3>以下の御注文を承りました</h3>
<#slot>
<br>
<input type=submit action="next" label="最初に戻る">
```

第 8 図

cgios

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

戻る 進む 停止 検索 お気に入り メディア 移動

アドレス(D) http://OOOO.net/oyster_r2/sp/kernel

牡蛎の銘柄・調理方法を選択してください

牡蛎の銘柄 岩牡蛎

牡蛎の調理方法 生食

添える調味料 ☒ レモン ☐ カクテルソース ☐ ホースラディッシュ

コメント

次へ

ページが表示されました

インターネット

第 9 図

cgios

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

戻る 進む 停止 検索 お気に入り メディア 移動

アドレス(D) http://OOOO.net/oyster_r2/sp/kernel

牡蛎の銘柄・調理方法を選択してください

牡蛎の銘柄 岩牡蛎

牡蛎の調理方法 生食

添える調味料 ☒ レモン ☒ カクテルソース ☐ ホースラディッシュ

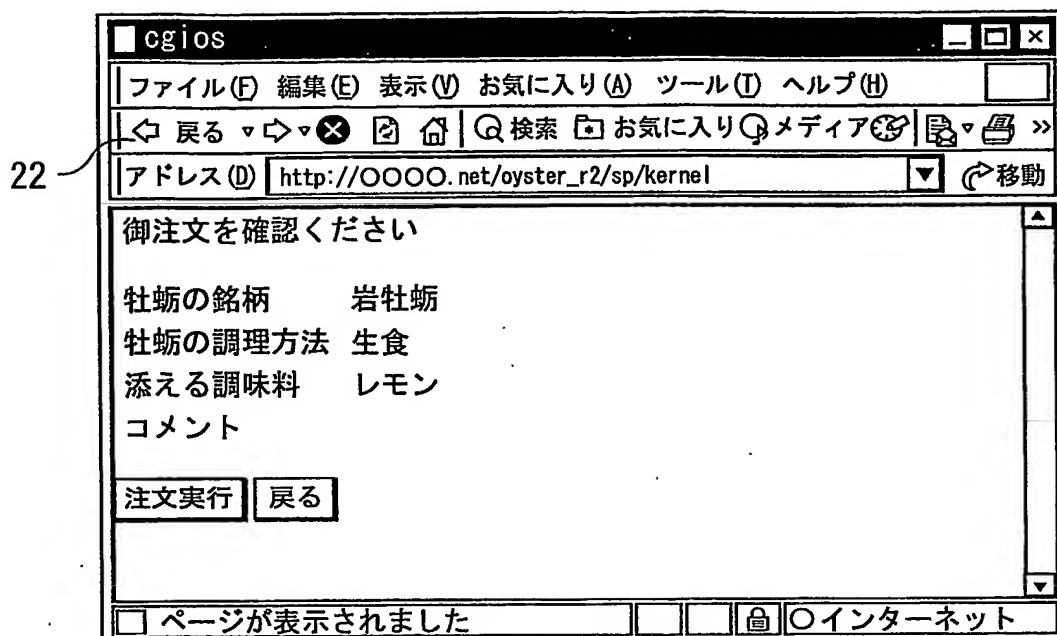
コメント

次へ

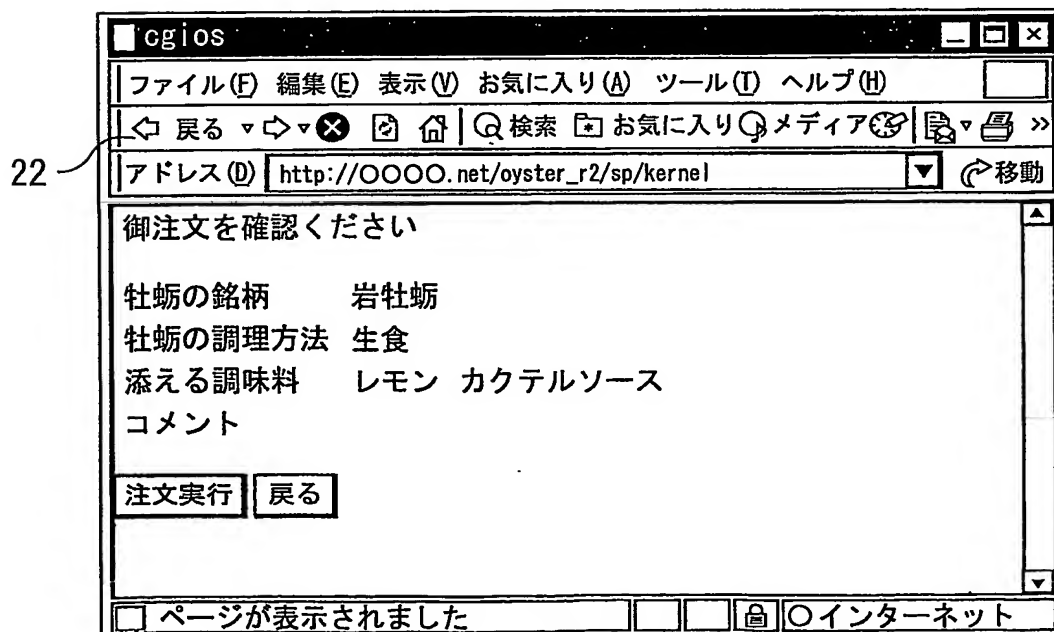
ページが表示されました

インターネット

第 10 図



第 11 図



第 1 2 図

22

cgios

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

戻る 進む 印刷 検索 お気に入り メディア 移動

アドレス(D) http://OOOO.net/oyster_r2/sp/kernel

牡蛎の銘柄・調理方法を選択してください

・同時に添付できる調味料は2種類までです

牡蛎の銘柄 岩牡蛎

牡蛎の調理方法 生食

添える調味料 ☒ レモン ☒ カクテルソース ☒ ホースラディッシュ

コメント

次へ

☐ ページが表示されました ☐ ☐ インターネット

第 1 3 図

22

cgios

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

戻る 進む 印刷 検索 お気に入り メディア 移動

アドレス(D) http://OOOO.net/oyster_r2/sp/kernel

以下の御注文を承りました

牡蛎の銘柄 岩牡蛎

牡蛎の調理方法 生食

添える調味料 レモン

コメント

最初に戻る

☐ ページが表示されました ☐ ☐ インターネット

20

21

第14図

22

cgios

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

戻る 進む 検索 入力 検索 お気に入り メディア 移動

アドレス(D) http://OOOO.net/oyster_r2/sp/kernel

牡蛎の銘柄・調理方法を選択してください

20

牡蛎の銘柄 岩牡蛎

牡蛎の調理方法 生食

添える調味料 ☒ レモン ☐ カクテルソース ☐ ホースラディッシュ

コメント

20

牡蛎の銘柄 岩牡蛎

牡蛎の調理方法 生食

添える調味料 ☒ レモン ☐ カクテルソース ☐ ホースラディッシュ

コメント

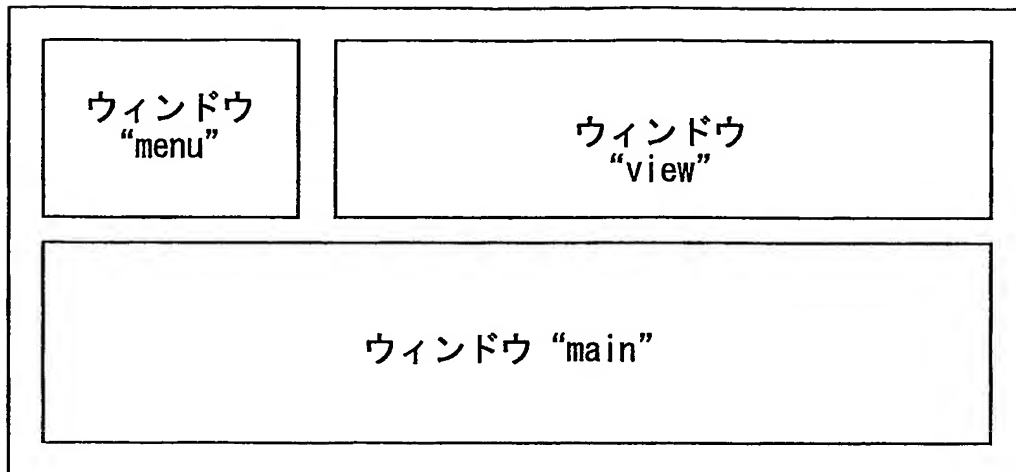
21

次へ

ページが表示されました

インターネット

第 15 図



フレーム “/main/top.page”

第 18 図

第16図

(a) <起動ストリーム:/init.stream>

```
ustream init() { # 起動ストリーム
  loop:
    p_fork("ウィンドウストリーム","/main/init.stream","this");
    p_wait();
    goto loop;
}
```

(b) <ウィンドウ起動ストリーム:/main/init.stream>

```
ustream init() { # ウィンドウストリーム
  loop:
    p_setPage("top.page");
    p_fork("メニュー","menu.stream","menu");
    p_wait();
    goto loop;
}
```

(c) <メニューストリーム:/main/menu.stream>

```
ustream menu() { # メニュー表示ストリーム
  first:
    page(menu.page) {
    }
    if ($V->{action} eq "input_order") {
      p_fork("注文入力","input_order.stream","^view");
    } elsif ($V->{action} eq "list") {
      p_fork("注文一覧表示","list.stream","^main");
    }
    goto first;
}
```


第17図

(a) <トップフレーム(新規追加):/main/top.page>

```

<table width=100% height=100% bgcolor=#ffffff border=0 cellspacing=2
cellpadding=0 style="table-layout:fixed">
  <tr>
    <td width=143 valign=top align=left class=wallpaper>
      <div style="overflow:auto"><#menu></div></td>
    <td valign=top class=wallpaper>
      <div style="overflow:auto"><#view></div></td>
  </tr>
  <tr>
    <td colspan=2 height=70% valign=top class=wallpaper>
      <div style="overflow:auto"><#main></div></td>
  </tr>
</table>

```

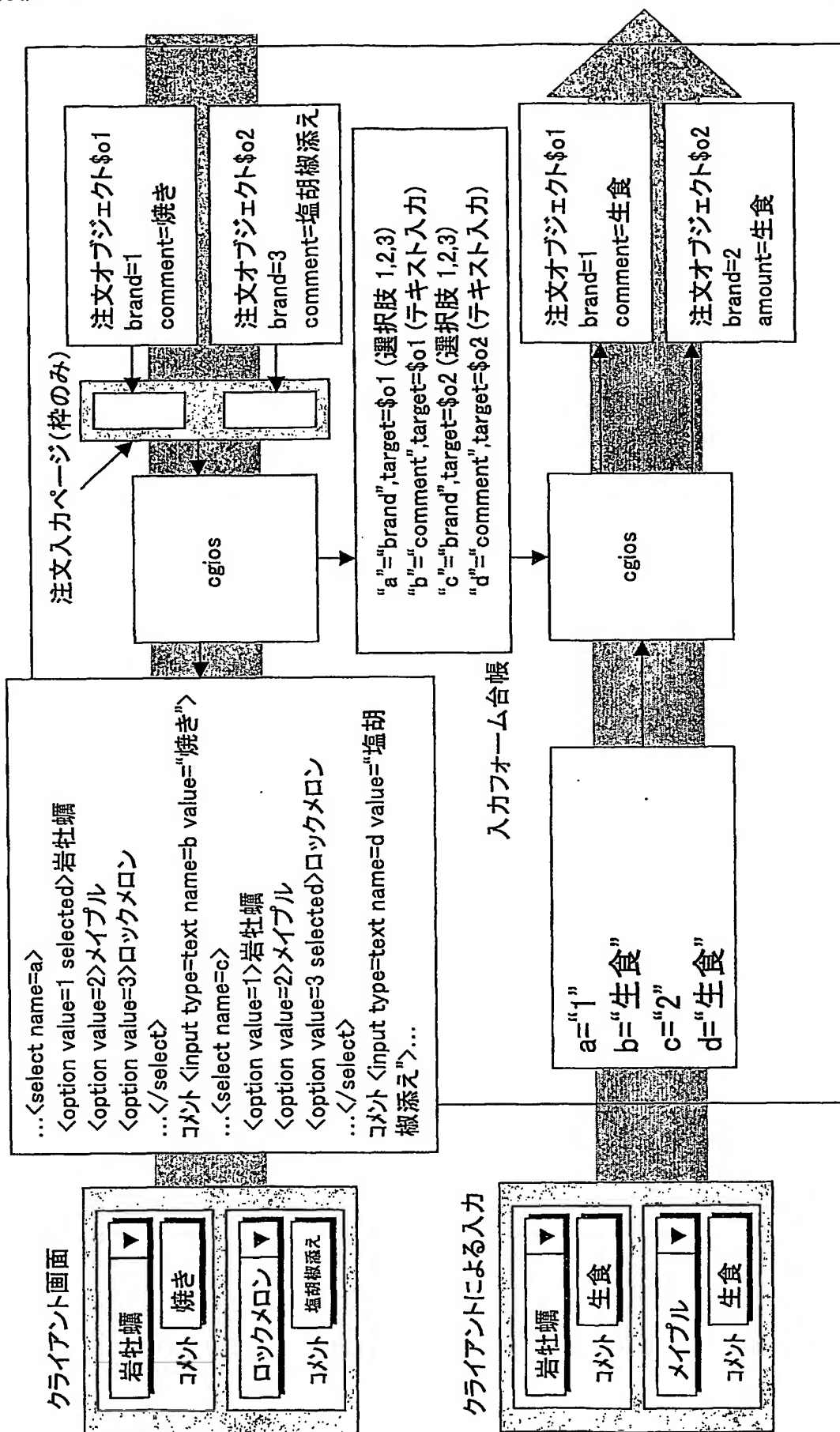
(b) <メニューページ(新規追加):/main/menu.page>

```

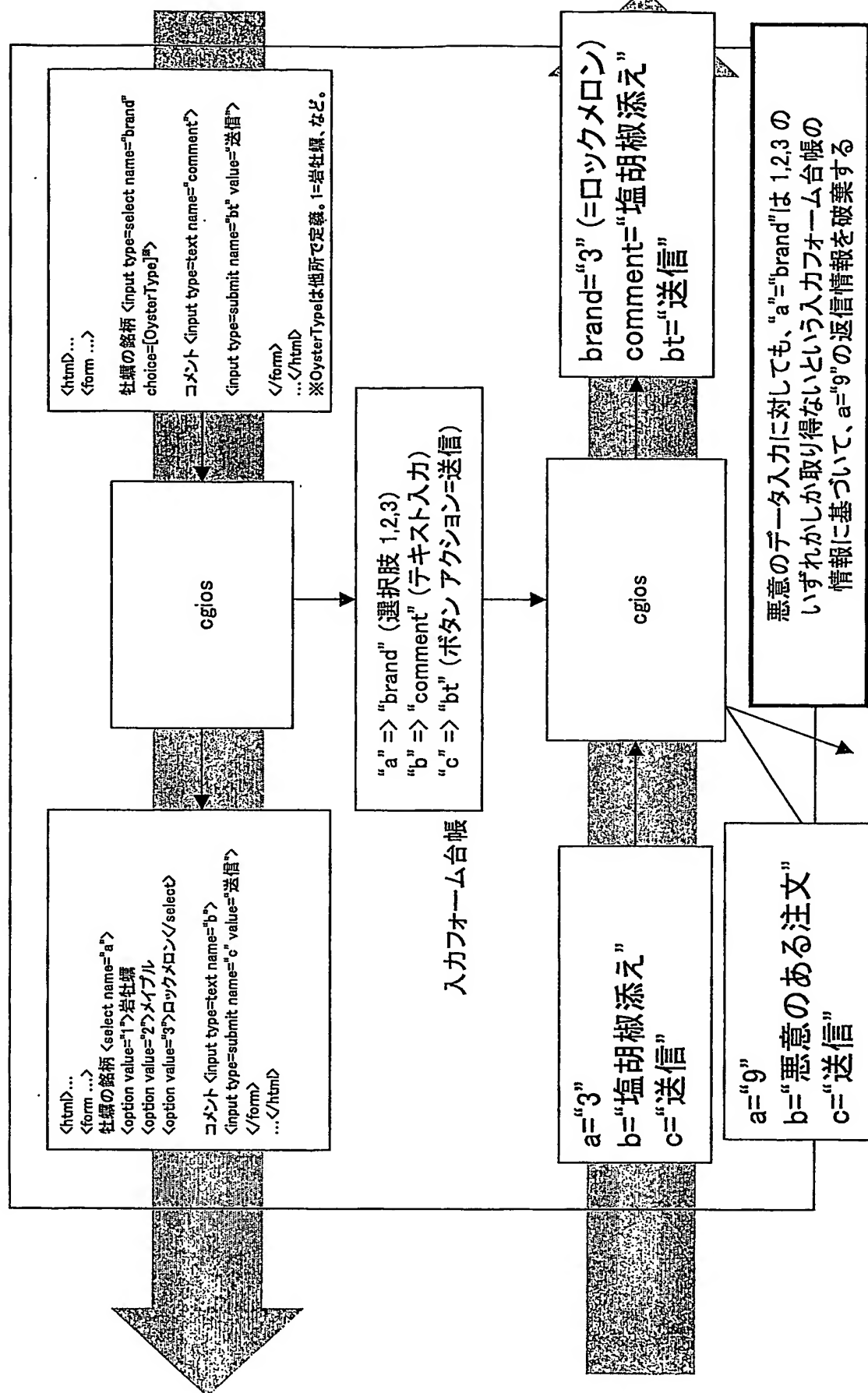
<h3>メインメニュー</h3>
<table><tr><td valign=top>
  <table class=inner bgcolor=#ffffff border-color=#000000 border=2
cellspacing=0 cellpadding=10>
    <tr><td valign=top>
      <li><a href="action:input_order">注文入力</a>
      <li><a href="action:list">注文一覧表示</a>
    </td></tr>
  </table>
</td></tr></table>

```

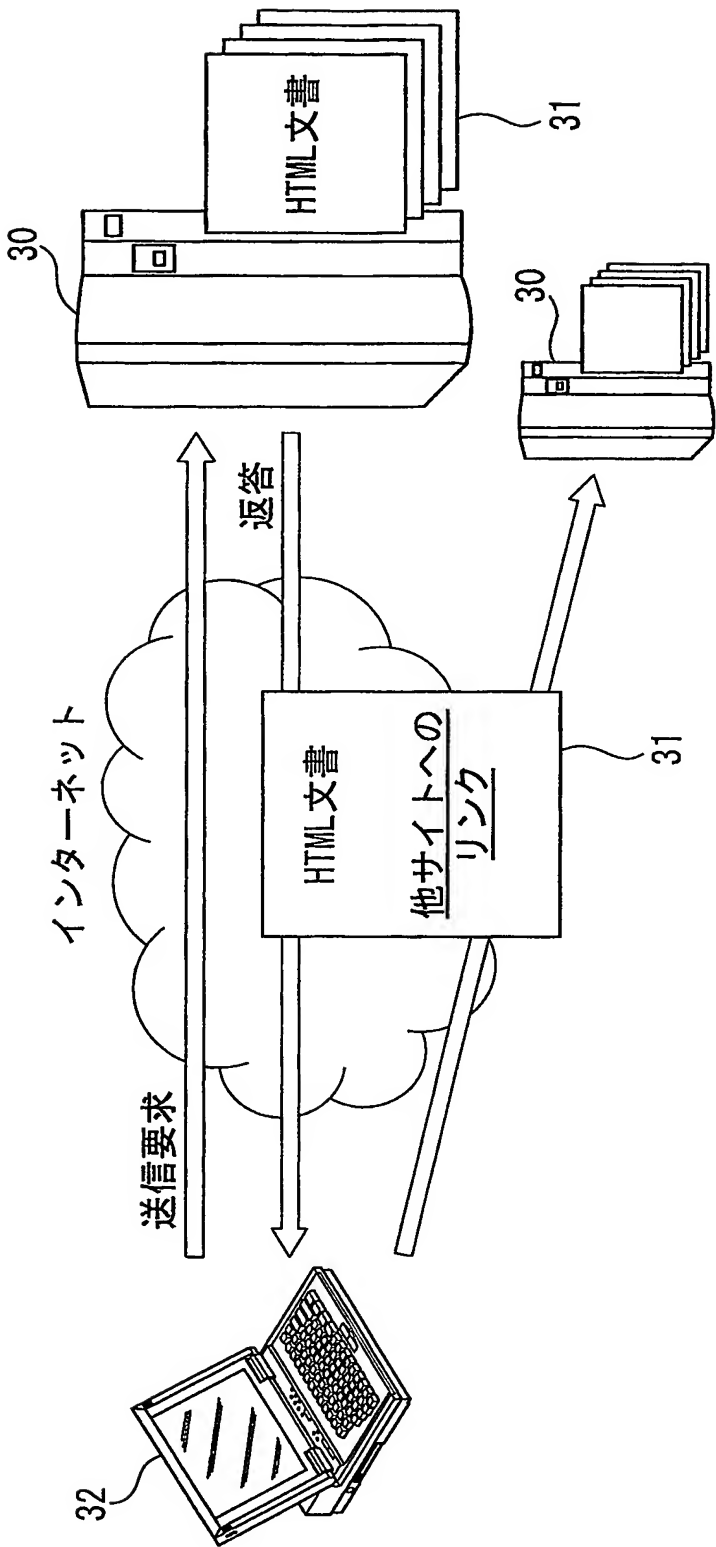
第 19 図



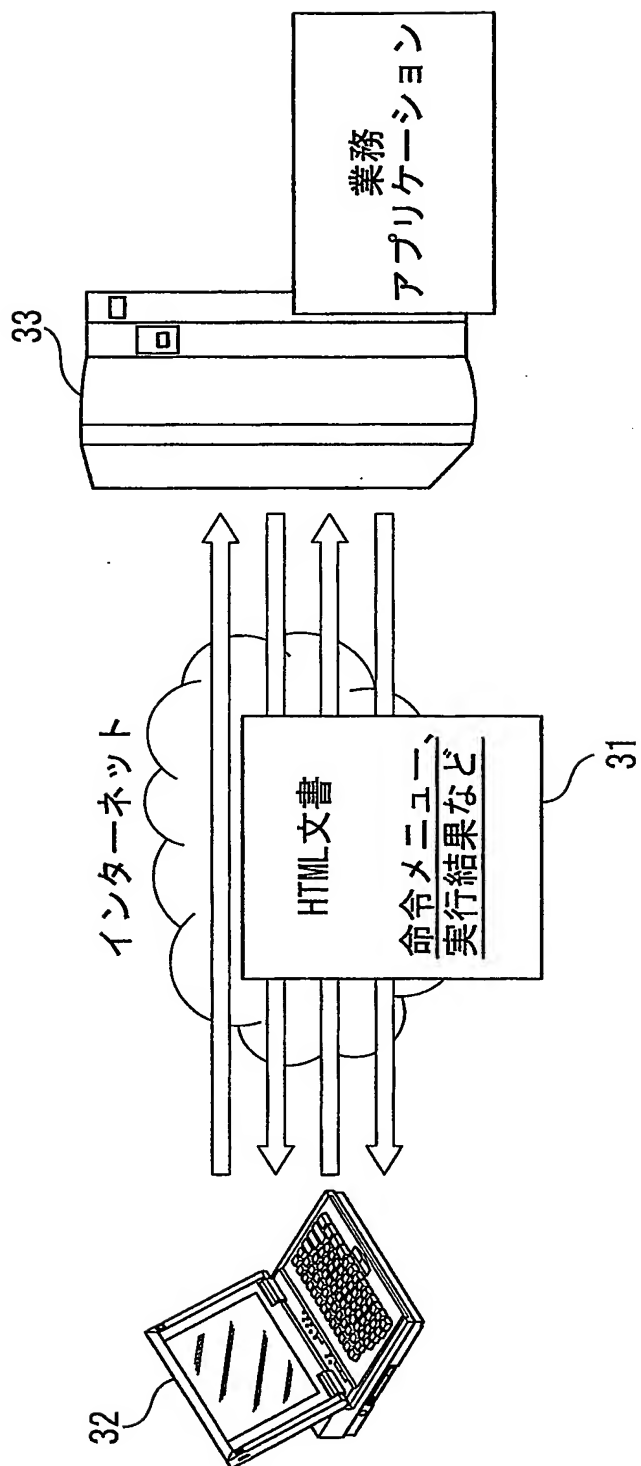
第20図



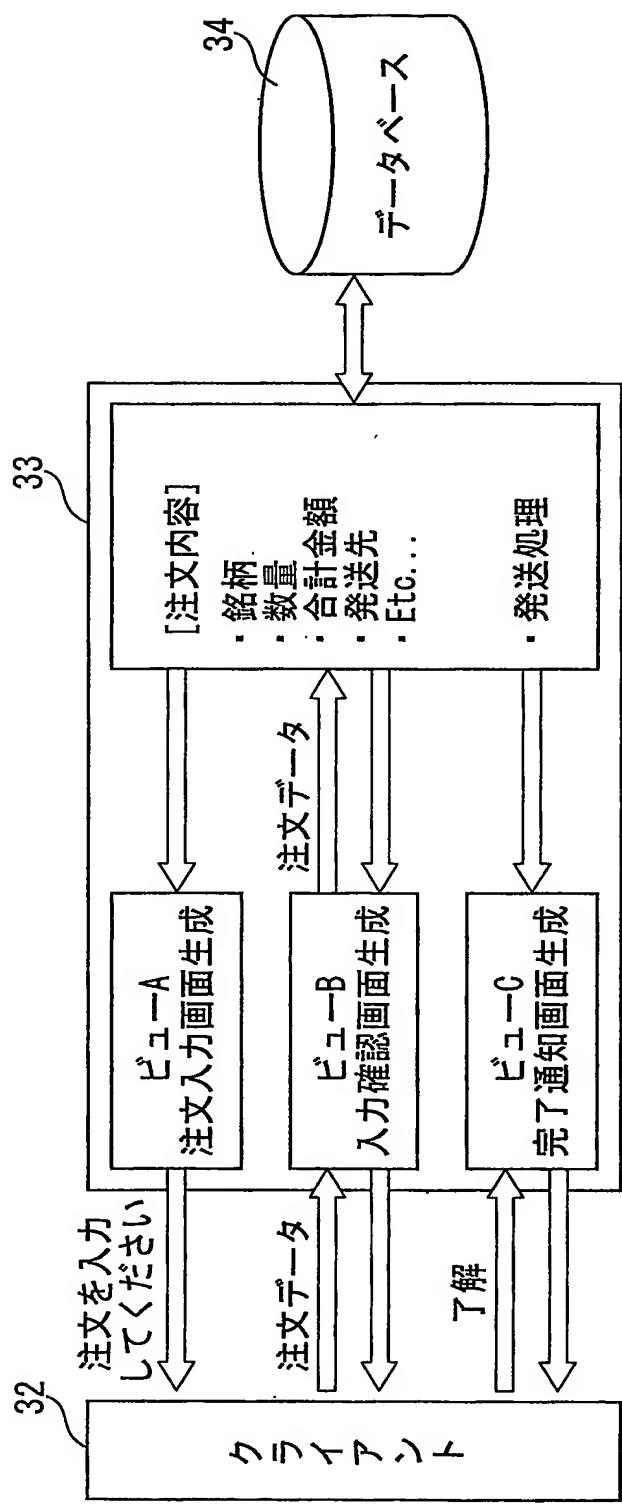
第 2 2 図



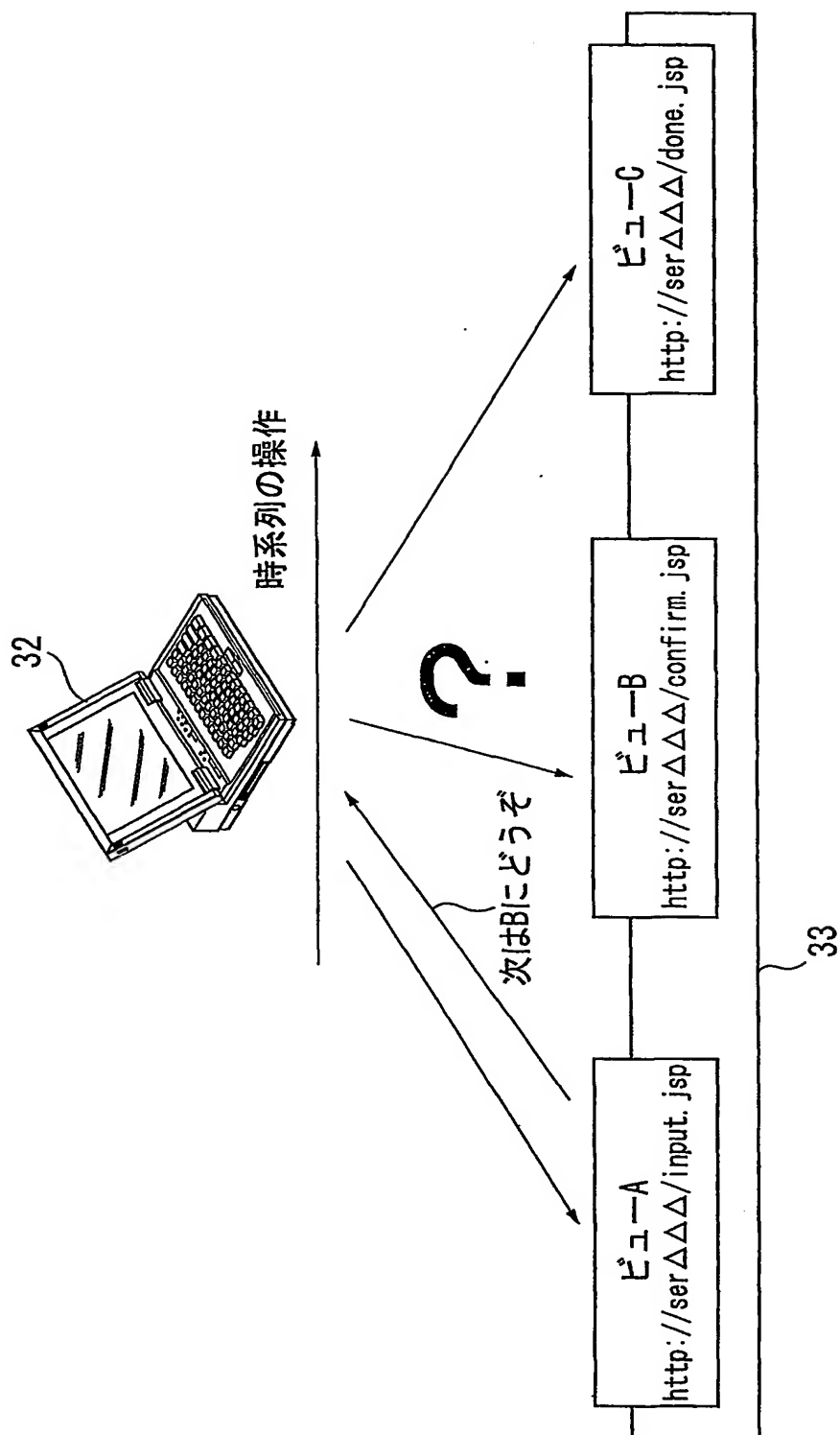
第 2 3 図



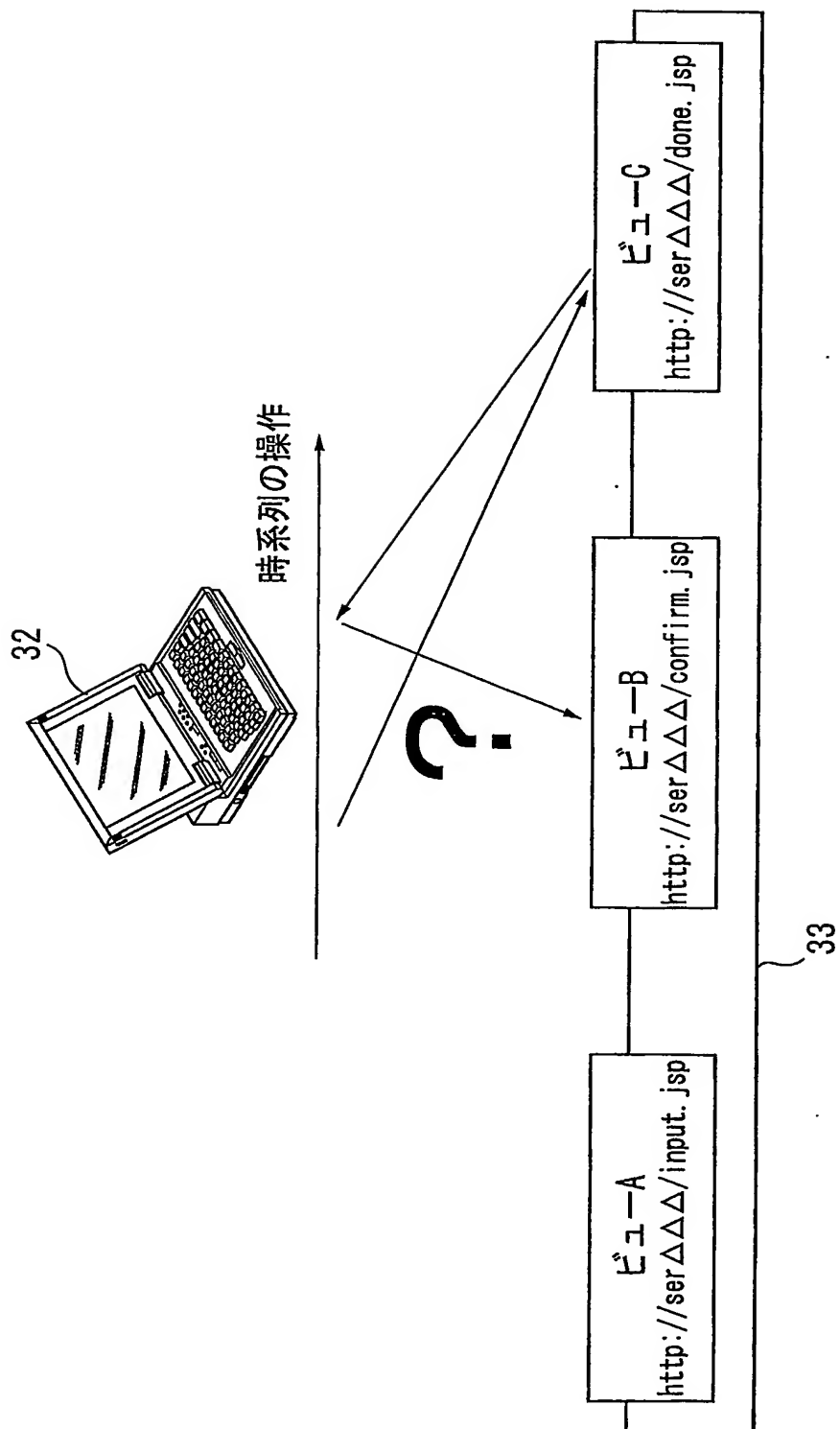
第24図



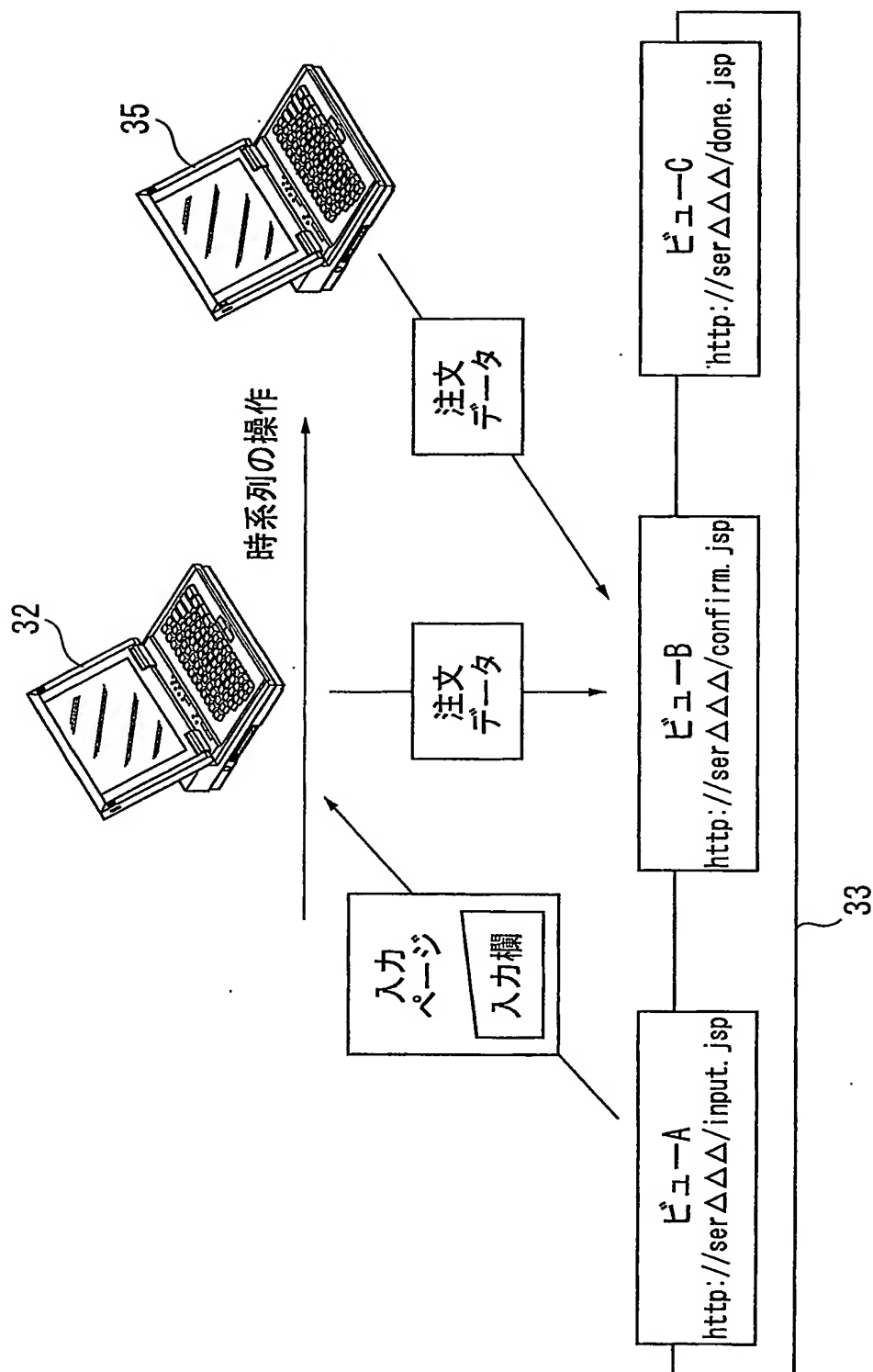
第25図



第26図



第 2 7 図



INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP03/16231

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl⁷ G06F15/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl⁷ G06F15/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1922-1996 Toroku Jitsuyo Shinan Koho 1994-2004
Kokai Jitsuyo Shinan Koho 1971-2004 Jitsuyo Shinan Toroku Koho 1996-2004

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 10-257048 A (International Business Machines Corp.), 25 September, 1998 (25.09.98), Full text; Figs. 1 to 4 & US 5875296 B	1-10
E,A	JP 2003-44396 A (Fujitsu Ltd.), 14 February, 2003 (14.02.03), Full text; Figs. 1 to 8 (Family: none)	1-10
A	JP 2002-236662 A (NEC Corp.), 23 August, 2002 (23.08.02), Full text; Figs. 1 to 4 (Family: none)	1-10

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family
---	--

Date of the actual completion of the international search
30 January, 2004 (30.01.04)

Date of mailing of the international search report
10 February, 2004 (10.02.04)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP03/16231

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2002-334057 A (Hitachi, Ltd.), 22 November, 2002 (22.11.02), Full text; Figs. 1 to 4 (Family: none)	1-10

A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int. Cl. G06F15/00

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int. Cl. G06F15/00

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報 1922-1996年
 日本国公開実用新案公報 1971-2004年
 日本国登録実用新案公報 1994-2004年
 日本国実用新案登録公報 1996-2004年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
A	JP 10-257048 A (インターナショナル・ビジネス・マシーンズ・コーポレーション) 1998.09.25, 全文, 第1-4図 & US 5875296 B	1-10
EA	JP 2003-44396 A (富士通株式会社) 2003.02.14, 全文, 第1-8図 (ファミリーなし)	1-10

☒ C欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」 特に関連のある文献ではなく、一般的技術水準を示すもの

「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)

「O」 口頭による開示、使用、展示等に言及する文献

「P」 国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」 同一パテントファミリー文献

国際調査を完了した日

30.01.2004

国際調査報告の発送日

10.2.2004

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JP)

郵便番号100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

後藤 和茂

5B

9463

電話番号 03-3581-1101 内線 6907

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
A	J P 2 0 0 2 - 2 3 6 6 6 2 A (日本電気株式会社) 2 0 0 2 . 0 8 . 2 3 , 全文, 第1-4図 (ファミリーなし)	1-10
A	J P 2 0 0 2 - 3 3 4 0 5 7 A (株式会社日立製作所) 2 0 0 2 . 1 1 . 2 2 , 全文, 第1-4図 (ファミリーなし)	1-10

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☒ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.